Timely Inference over Networks

by

Md Kamran Chowdhury Shisher

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 4, 2024

Keywords: remote inference, age of information, scheduling, restless multi-armed bandit

Approved by

Yin Sun, Associate Professor of Electrical and Computer Engineering, Auburn University
Roy Yates, Distinguished Professor Emeritus, Electrical and Computer Engineering, Rutgers University
John Hung, Professor Emeritus of Electrical and Computer Engineering, Auburn University
Shiwen Mao, Professor of Electrical and Computer Engineering, Auburn University
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering, Auburn University
Tao Shu, Associate Professor of Computer Science and Software Engineering

Abstract

Next-generation communications (Next-G), such as 6G, are expected to support emerging networked intelligent systems, including autonomous driving, factory automation, digital twin technology, unmanned aerial vehicle (UAV) navigation, and extended reality. Timely inference is crucial for these networked intelligent systems. In this dissertation, we investigate a remote inference system, where a trained neural network is used to infer time-varying targets (e.g., the locations of vehicles and pedestrians) based on features (e.g., video frames) observed at a sensing node (e.g., a camera). The inference error is determined by (i) the timeliness and (ii) the sequence length of the features, where we use the Age of Information (AoI) as a metric for timeliness.

In the first part of the dissertation, we discuss how to evaluate the importance of timely information in remote inference and the monotonicity of information aging. One might expect that the performance of a remote inference system degrades monotonically as the feature becomes stale. Using a new information-theoretic analysis, we show that this is true if the feature and target data sequence can be closely approximated as a Markov chain; it is not true if the data sequence is far from Markovian. Hence, the inference error is a function of the AoI, where the function could be non-monotonic. In addition, a longer feature can typically provide better inference performance, but it often requires more channel resources for sending the feature.

In the second part of the dissertation, we study the transmission scheduling problem that optimizes timeliness and feature sequence length to minimize inference error. We introduce a new "selection-from-buffer" medium access model for status updating and minimize inference errors for both Markovian and non-Markovian data. For single-source and single-channel remote inference networks, we obtain optimal scheduling policies for both the cases of time-invariant and time-variant feature lengths. In multi-sources and multi-channel remote inference networks, the selection-from-buffer scheduling problem is a multi-action restless multi-arm bandit problem. For this setting, we design new scheduling policies by utilizing the *Whittle index* and duality-based *gain index*. The new scheduling policies are proven to be

asymptotically optimal. Data-driven evaluations show that our policies can reduce inference errors by up to 10,000 times.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Prof. Yin Sun, who provided me with helpful guidance and continuous support during my Ph.D. studies. His constant encouragement, serious attitude, and enthusiasm towards good research have motivated me to do my best. He has been so helpful and considerate in my life. I feel grateful and fortunate to have him as an advisor.

I would also like to thank my dissertation committee: Prof. Roy Yates, Prof. John Hung, Prof. Shiwen Mao, and Prof. Thaddeus Roppel, for their valuable comments and generous support. I am also indebted to Prof. Tao Shu for serving as the university reader and reviewing my work. Additionally, I express my gratitude to my collaborators: Prof. I-Hong Hou, Prof. Bo Ji, Prof. Elif Uysal, Prof. Lei Yang, Prof. Feng Yan, Dr. Clement Kam, Dr. Heyang Qin, Tasmeen Zaman Ornee, Cagri Ari, and Kevin Yan.

I want to express my thankfulness to my current and former colleagues: Dr. Jiayu Pan, Tasmeen Zaman Ornee, Shaoyi Li, and Sam Chamoun for their feedback and technical discussions. I also thank all my friends in Auburn for the pleasant memories and friendship, which will be treasured for the rest of my life. Moreover, I want to thank my younger brother, Md Sadman Chowdhury, and my younger sister, Noshin Atia Chowdhury Sneha, for their love, prayers, and support.

I would like to extend my heartfelt thanks to my mother, Mrs. Kamrunnahar Shimul, and my father, Md. Golam Mahfuz Chowdhury, for their continuous support and tremendous care in my life, without whom I wouldn't have made it to this day. Finally, I want to express my deepest gratitude to my wife, Tasmeen Zaman Ornee, who accompanied me throughout my Ph.D. journey. Without her patience, support, faith, and care, it would not have been possible for me to complete this journey.

This work was supported in part by the NSF grant CCF-1813050, CNS-2239677, and CCF-1813078; the ARO grant W911NF-21-1-0244; and the ONR grant N00014-17-1-2417.

Table of Contents

List of Figures

Chapter 1

Introduction

Next-generation communications (Next-G), such as 6G, are expected to support a wide range of emerging networked intelligent systems, including autonomous driving, factory automation, digital twins, UAV navigation, extended reality, and more [3–8]. Timely inference or estimation is crucial for these systems. For instance, an autonomous vehicle infers the trajectories of nearby vehicles and the intentions of pedestrians based on features collected from lidars and cameras installed on the vehicle. In remote surgery, the movement of a surgical robot is predicted in real-time. In this dissertation, we study a remote inference system as illustrated in Fig. 1.1, where a trained neural network infers a time-varying target (e.g., the location of a vehicle) based on features (e.g., video frames) sent from a sensing node (e.g., camera). Due to communication delay and transmission errors, the features delivered to the neural network may not be fresh. Information freshness, in this context, refers to the time elapsed between feature generation and its use in inference. While not a concern in traditional offline inference, the impact of information freshness on time-sensitive inference systems remains an under-explored area of research. It is crucial to understand how this freshness affects inference performance.

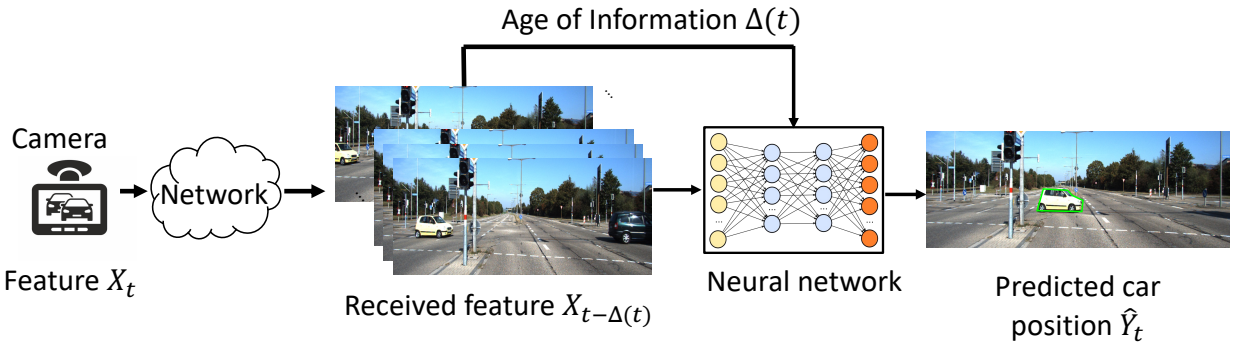


Figure 1.1: A remote inference system, where a trained neural network infers the location of a vehicle based on video frames sent from a camera.

In a communication system, understanding the freshness of information at the receiver is crucial. Imagine trying to make decisions about a drone flight using outdated location

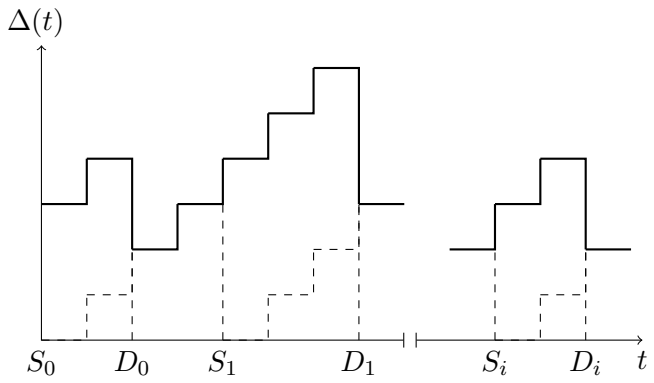


Figure 1.2: Evolution of the AoI $\Delta(t)$ in discrete-time.

data. The concept of Age of Information (AoI) was introduced to measure the information freshness of the receiver side of a communication system [9,10]. Formally, consider a sequence of status updates sent from a source to a receiver. Let $U(t)$ be the generation time of the freshest packet delivered by time t. The AoI, $\Delta(t)$, is the difference between the current time t and $U(t)$:

$$\Delta(t) := t - U(t). \tag{1.1}$$

To understand more about the evolution of AoI in a time-slotted system, see Fig. 1.2, where S_i and D_i are the generation time and delivered time of the i-th packet, respectively, such that $S_i < S_{i+1} < D_{i+1}$. Then, $U(t) = \max\{S_i : D_i \leq t\}$ is the generation time of the freshest packet that has been delivered to the receiver by time t. Then, from (1.1), we get

$$\Delta(t) = t - U(t) = t - \max\{S_i : D_i \leq t\}. \tag{1.2}$$

A smaller AoI indicates the presence of more recent information at the receiver.

The freshness of information is tightly related to its significance for the application. "Freshness" is an example of the "semantic" properties of the information. By semantics, we mean properties that relate to the "purpose" of information, which, in our case, is to improve the performance of remote inference systems using fresh information. In semantic-aware communications, the design goal is no longer sending as much as possible (i.e., increase

throughput), or as fast as possible (i.e., reduce delay), without regard what is the intended use of information. Instead, the transmitter should choose the right piece of information that is important for the system operating at the receiving end, and try to deliver the information to the receiver while it is still useful. However, little is known on how to measure the importance of information in a broad range of real-time systems and applications. Towards this end, in this dissertation, we answer two questions: (i) *How to evaluate the importance of fresh information in a remote inference/estimation system?* (ii) *How to improve the performance of a remote inference/estimation system?*

1.1 Outline and Main Contributions

In Chapter 2, we answer the first question: How to evaluate the importance of fresh information in a remote inference/estimation system?. This chapter is based on work presented first in papers [11–14]. One might assume that inference errors degrade monotonically as the data becomes stale. However, the remote inference experiments illustrated in Chapter 2 show that this assumption holds true in some scenarios (e.g., video prediction) and not in others (e.g., temperature prediction and reaction prediction). For example, in predicting next hour temperature, temperature recorded at 24 hours ago is more relevant than temperature recorded at 12 hours ago due to daily weather patterns. Moreover, consider an example of reaction prediction: If a vehicle initiates braking, nearby vehicles don't react instantly due to the response delay of the drivers or the braking systems. Therefore, slightly outdated actions can be more relevant for predicting reactions than the most current action. These observations highlight that fresher data is not always better. To accurately assess the value of fresh information, we need a robust analytical framework. In this chapter, we present a new information-theoretic analysis with two analytical models to reveal when fresher data is always better and when it is not.

Next, we focus on the design of communication systems to improve the performance of a remote inference system. Due to the vast amount of data generated by today's sensing nodes (e.g. cameras and lidars installed on vehicles or roadside units) and the constraints of limited communication resources, relying solely on maximizing data transmission for ensuring timely and accurate performance is inadequate. The unprecedented data volume necessitates

a paradigm shift in communication strategies—from maximizing data transmission rates to extracting and transmitting the right piece of information (often referred to as *semantic information*) to accomplish specific tasks such as accurate inference (e.g., the location of a vehicle) [15–19]. This communication paradigm is known as semantic communication.

In Chapter 3, we investigate how to design a semantic communication system by considering "Freshness" as a "semantic" property of the information. This chapter builds upon the work presented in [12, 14]. Specifically, we study single-source single-channel scheduling problems to reduce inference error. As found in Chapter 2, inference error can be expressed a function of AoI, but not necessarily a monotonic function of AoI. Most existing studies in AoI literature [20–32] have focused on designing transmission scheduling strategies to minimize monotonic functions of AoI. However, the design of efficient scheduling policies for optimizing general, potentially non-monotonic functions of AoI remains unexplored. We design transmission scheduling policies for single-source and single channel system to minimize the inference error and general functions of AoI, regardless of whether the functions are monotonic or non-monotonic.

In many networked intelligent systems, a receiver (e.g., an autonomous vehicle) requires information from multiple sources (e.g., onboard cameras, roadside units, and nearby vehicles). Due to limited communication resources, it is impractical to obtain information from all sources simultaneously. A scheduler decides which sources to select and what information from sources to send to accomplish a task. Hence, a follow-up question is *how to design a scheduler for multi-source multi-channel remote inference systems?*

Based on the work in [12, 14], we design a scheduler for multi-source multi-channel remote inference systems in Chapter 4. When there are multiple source-predictor pairs and multiple channels, the scheduling problem is a restless multi-armed bandit (RMAB) problem with multiple actions. Whittle index policy is a well-known asymptotically optimal policy for RMAB problems with binary actions [33]. However, Whittle index policy is not sufficient for problems with multiple actions. We propose a multi-source, multi-action scheduling policy that uses a Whittle index algorithm to determine which sources to schedule and employs a duality-based action selection algorithm to decide which information to send from the selected sources.

Additionally, the performance of remote inference depends on the sequence length of features. Longer feature sequences can carry more information about the target, resulting in the reduction of inference errors. Though a longer feature can provide better training and inference performance, it often requires more communication resources. For example, a longer feature may require a longer transmission time and may end up being stale when delivered, thus resulting in worse inference performance. Hence, it is necessary to study a learning and communications co-design problem that jointly controls the timeliness and the length of the feature sequences.

Based on the work in [34], we study a learning and communication co-design problem that jointly optimizes feature length selection and transmission scheduling to minimize the time-averaged inference error in Chapter 5. When there is a single source-predictor pair and a single channel, we develop low-complexity optimal co-designs for both the cases of time-invariant and time-variant feature length. When there are multiple source-predictor pairs and multiple channels, the co-design problem becomes a RMAB problem with multiple actions. For this setting, we can not use Whittle index because it is difficult to establish Whittle indexability for time-varying feature length. To solve the problem, we design a low-complexity scheduling algorithm that does not require to satisfy any indexability condition.

Numerical results in this dissertation demonstrate that potential of our scheduling algorithms to reduce inference error by up to 10000 times.

Chapter 2

Interpretation of Information Freshness in Remote Inference

2.1 Introduction

In this chapter, we first examine the importance of information freshness on remote inference. While one might assume that inference errors degrade monotonically as the data becomes stale, the remote inference experiments illustrated in this chapter show that this assumption holds true in some scenarios (e.g., video prediction) and not in others (e.g., temperature prediction and reaction prediction). By using an information-theoretic analysis, we reveal that inference errors are functions of the AoI, whereas the function is not necessarily monotonic.

The contributions of this chapter are summarized as follows:

- We conduct five experiments to examine the impact of data freshness on remote inference (see Section 2.3). These experiments include (i) video prediction, (ii) robot state prediction in a leader-follower robotic system, (iii) actuator state prediction under mechanical response delay, (iv) temperature prediction, and (v) wireless channel state information (CSI) prediction. One might assume that the inference error degrades monotonically as the data becomes stale. Our experimental results show that this assumption is not always true. In some scenarios, even the fresh data with $\Delta(t) = 0$ may generate a larger inference error than stale data with $\Delta(t) > 0$ (see Figs 2.2-2.3).

- We develop an information-theoretical method to interpret these counter-intuitive experimental results (see Section 2.5). By introducing a new local information geometric tool, we show that the assumption "fresh data is better than stale data" is true when the time-sequence data used for remote inference can be closely approximated as a Markov chain; but it is not true when the data sequence is far from Markovian (Theorems 2.1-2.3). Specifically, we proposed a new tool called *ϵ-Markov chain*. By using *ϵ-Markov chain*, we can evaluate the divergence of a target process from being a Markov chain. When the divergence ϵ is large, the inference error can be far from a

non-decreasing function of AoI. Conversely, for small divergence ϵ, the estimation error is close to a non-decreasing AoI function. Hence, the inference error is a function of the AoI, whereas the function is not necessarily monotonic. This analysis provides an information-theoretic interpretation of information freshness.

- Moreover, we construct two analytical models to analyze and explain when fresh data is better than stale data and when it is not (see Section 2.6). We derive closed-form expressions for the remote estimation error of Gaussian autoregressive AR(p) processes and study how the monotonicity of information aging is affected by the parameters of the AR(p) process.

2.1.1 Related Works

The concept of Age of Information (AoI) has attracted significant research interest; see, e.g., [10, 20–32, 35–42] and a recent survey [43]. Initially, research efforts were centered on analyzing and optimizing average AoI and peak AoI in communication networks [10, 20, 21, 29,35]. Recent research endeavors on AoI have shifted towards optimizing the performance of real-time applications, such as remote estimation, remote inference, and control systems, by leveraging AoI as a tool. In [23, 24, 36, 38], information-theoretic metrics such as Shannon's mutual information (or Shannon's conditional entropy) has been used to quantify the amount of information carried by the received data about the current source value (or the amount of uncertainty regarding the current source value) as the data ages. In addition, a Shannon's conditional entropy term $H_{\text{Shannon}}(Y_t|X_{t-\Delta(t)} = x, \Delta(t) = \delta)$ was used in [37] to quantify information uncertainty given the most recent observation $X_{t-\Delta(t)} = x$ and AoI $\Delta(t) = \delta$. The information-theoretic metrics in these prior studies [23, 24, 36–38] cannot be directly used to evaluate system performance. To bridge the gap, we use an L-conditional entropy $H_L(Y_t|X_{t-\Delta(t)}, \Delta(t))$, to approximate and analyze the inference error in remote inference, as well as the estimation error in remote estimation [11, 12, 14, 34]. For example, when the loss function $L(y, \hat{y})$ is chosen as a quadratic function $||y - \hat{y}||_2^2$, the L-conditional entropy $H_L(Y_t|X_{t-\Delta(t)}, \Delta(t)) = \mathbb{E}[(Y_t - \mathbb{E}[Y_t|X_{t-\Delta(t)}, \Delta(t)])^2]$ is exactly the minimum mean squared estimation error in signal-agnostic remote estimation. This approach takes a significant step

to bridge the gap between AoI metrics and real-world applications, by directly mapping the AoI to the application performance metrics.

This study is also related to the field of signal-agnostic remote estimation. The prior studies [23, 25, 27, 39, 40, 44, 45] in signal-agnostic remote estimation focused on Gaussian and Markovian processes. The results presented in the current paper are applicable to more general processes.

2.2 System Model

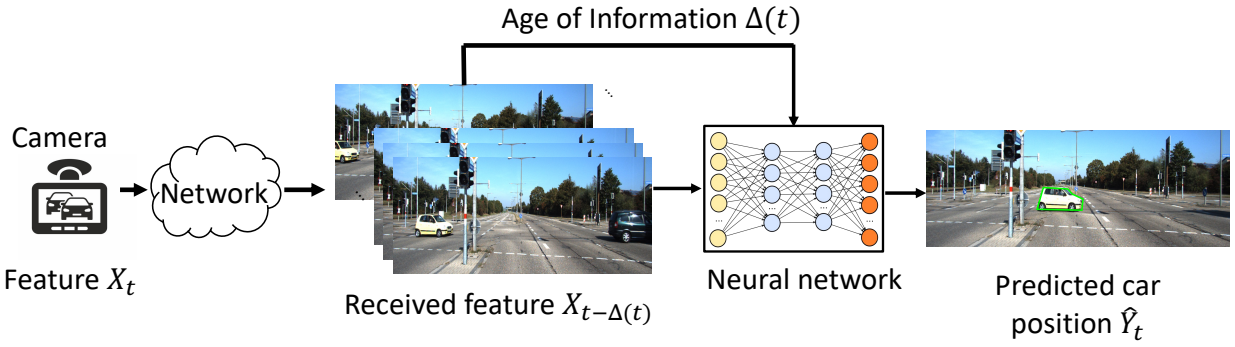
In this section, we introduce a remote inference system and illustrate the impact of information freshness on the inference performance.

Remote Inference Model

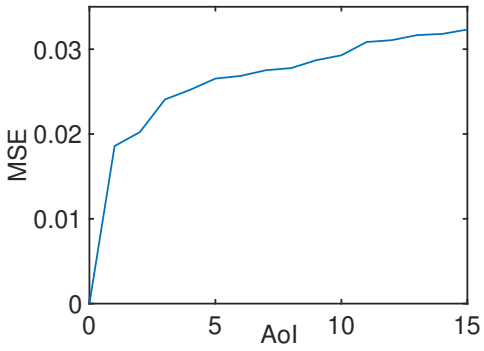
Consider the remote inference system illustrated in Fig. 2.1(a). In this system, a time-varying target $Y_t \in \mathcal{Y}$ (e.g., the position of the car in front) is predicted at time t, using a feature $X_{t-\Delta(t)} \in \mathcal{X}$ (e.g., a video clip) that was generated $\Delta(t)$ seconds ago at a sensor (e.g., a camera). The time difference $\Delta(t)$ between $X_{t-\Delta(t)}$ and Y_t is the AoI defined in (1.1). Each feature $X_t = (V_t, V_{t-1}, \ldots, V_{t-u+1})$ is a time series of length u, extracted from the sensor's output signal V_t. For example, if V_t is the video frame at time t, then X_t represents a video clip consisting of u consecutive video frames.

We focus on a class of popular supervised learning algorithms known as *Empirical Risk Minimization (ERM)* [46]. In freshness-aware ERM supervised learning algorithms, a neural network is trained to generate an action $a = \phi(X_{t-\Delta(t)}, \Delta(t)) \in \mathcal{A}$, where $\phi : \mathcal{X} \times \mathbb{Z}^+ \mapsto \mathcal{A}$ is a function that maps a feature $X_{t-\Delta(t)} \in \mathcal{X}$ and its AoI $\Delta(t) \in \mathbb{Z}^+$ to an action $a \in \mathcal{A}$. The performance of learning is evaluated using a loss function $L : \mathcal{Y} \times \mathcal{A} \mapsto \mathbb{R}$, where $L(y, a)$ represents the loss incurred if action a is selected when $Y_t = y$.

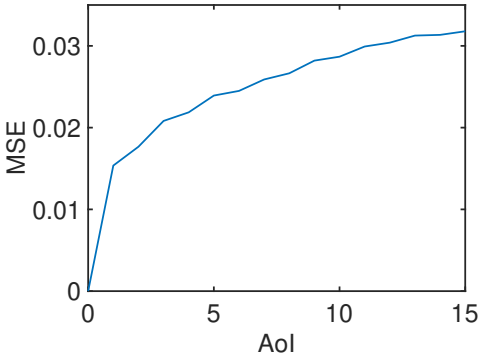
The loss function L is determined by the *goal* of the remote inference system. For example, in neural network-based minimum mean-squared estimation, the loss function is $L_2(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$, where the action $a = \hat{\mathbf{y}}$ is an estimate of the target $Y_t = \mathbf{y}$ and $\|\mathbf{y}\|_2^2$ is the Euclidean norm of the vector $\mathbf{y}$. In softmax regression (i.e., neural network-based



(a) Video prediction Task



(b) Training Error vs. AoI



(c) Inference Error vs. AoI

Figure 2.1: Performance of video prediction experiment. The experimental results in (b) and (c) are regenerated from [1]. The training and inference errors are non-decreasing functions of the AoI.

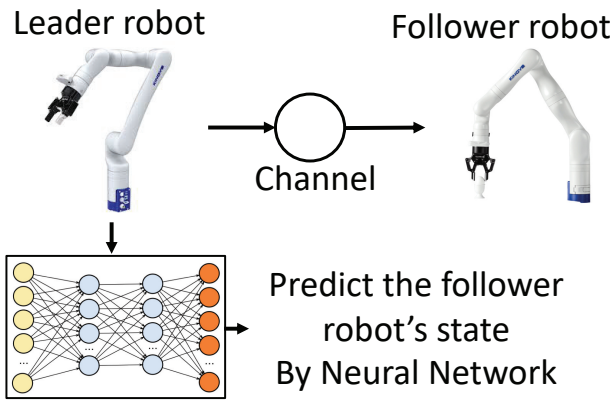
maximum likelihood classification), the action $a = Q_Y$ is a distribution of Y_t and the loss function $L_{\log}(y, Q_Y) = -\log Q_Y(y)$ is the negative log-likelihood of the target value $Y_t = y$.

Offline Training and Online Inference

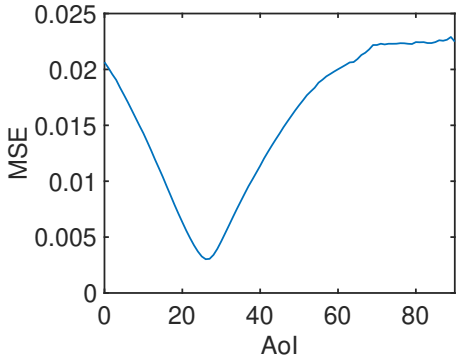
A supervised learning algorithm consists of two phases: *offline training* and *online inference*. In the offline training phase, a neural network based predictor is trained using one of the following two approaches.

In the first approach, multiple neural networks are trained independently, using distinct training datasets with different AoI values. The neural network associated with an AoI value δ is trained by solving the following ERM problem:

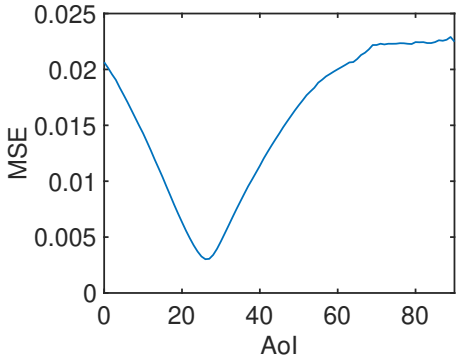
$$\text{err}_{\text{training},1}(\delta) = \min_{\phi \in \Lambda} \mathbb{E}_{Y,X \sim P_{\tilde{Y}_0, \tilde{X}_{-\delta}}}[L(Y, \phi(X, \delta))], \tag{2.1}$$



(a) Robot Prediction



(b) Training Error vs. AoI



(c) Inference Error vs. AoI

Figure 2.2: Robot state prediction in a leader-follower robotic system. The leader robot uses a neural network to predict the follower robot's state Y_t by using the leader robot's state $X_{t-\delta}$ generated δ time slots ago ($u = 1$). The training and inference errors decrease in the AoI ≤ 25 and increase when AoI ≥ 25.

where $P_{\tilde{Y}_0,\tilde{X}_{-\delta}}$ is the empirical distribution of the label $\tilde{Y}_0$ and the feature $\tilde{X}_{-\delta}$ in the training dataset, the AoI value δ is the time difference between $\tilde{Y}_0$ and $\tilde{X}_{-\delta}$, and Λ is the set of functions that can be constructed by the neural network.

In the second approach, a single neural network is trained using a larger dataset that encompasses a variety of AoI values. The ERM training problem for this approach is formulated as

$$\text{err}_{\text{training},2} = \min_{\phi\in\Lambda} \mathbb{E}_{Y,X,\Theta\sim P_{\tilde{Y}_0,\tilde{X}_{-\Theta},\Theta}}[L(Y,\phi(X,\Theta))], \tag{2.2}$$

where $P_{\tilde{Y}_0,\tilde{X}_{-\Theta},\Theta}$ is the empirical distribution of the label $\tilde{Y}_0$, the feature $\tilde{X}_{-\Theta}$, and the AoI Θ within the training dataset, and the AoI Θ is the time difference between $\tilde{Y}_0$ and $\tilde{X}_{-\Theta}$.

In the online inference phase, the trained neural predictor is used to predict the target Y_t in real-time. We assume that the process $\{(Y_t, X_t), t = 0, 1, 2, \ldots\}$ is stationary and is independent of the AoI process $\{\Delta(t), t = 0, 1, 2, \ldots\}$. Under these assumptions, if $\Delta(t) = \delta$, the inference error at time t can be expressed as a function of the AoI value δ, i.e.,

$$\mathrm{err}_{\mathrm{inference}}(\delta) = \mathbb{E}_{Y,X \sim P_{Y_t, X_{t-\delta}}}[L(Y, \phi^*(X, \delta))], \tag{2.3}$$

where $P_{Y_t, X_{t-\delta}}$ is the distribution of the target Y_t and the feature $X_{t-\delta}$, and ϕ^* is the trained neural network. The proof of (2.3) is provided in Appendix 2.D. In Sections 3-4, to minimize inference error, we will develop signal-agnostic transmission scheduling policies in which scheduling decisions are determined without using the knowledge of the signal value of the observed process. If the transmission schedule is signal-agnostic, then $\{(Y_t, X_t), t = 0, 1, 2, \ldots\}$ is independent of the AoI process $\{\Delta(t), t = 0, 1, 2, \ldots\}$. In Sections 3-4, to minimize inference error, we will develop signal-agnostic transmission scheduling policies in which scheduling decisions are determined without using the knowledge of the signal value of the observed process. If the transmission schedule is signal-agnostic, then $\{(Y_t, X_t), t = 0, 1, 2, \ldots\}$ is independent of the AoI process $\{\Delta(t), t = 0, 1, 2, \ldots\}$.

2.3 Experimental Results on Information Freshness

We conduct five remote inference experiments to examine how the training error and the inference error vary as the AoI increases. These experiments include (i) video prediction, (ii) robot state prediction in a leader-follower robotic system, (iii) actuator state prediction under mechanical response delay, (iv) temperature prediction, and (v) wireless channel state information prediction. In these experiments, we consider the quadratic loss function $L(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$. Detailed settings of these experiments can be found in Appendix 2.A. We present the experimental results of the first training method in Figs. 2.1-2.5. Related codes and datasets are accessible in our GitHub repository.[1] To illustrate the training error of the second training method as a function of the AoI δ, one can simply assess the training

[1] `https://github.com/Kamran0153/Impact-of-Data-Freshness-in-Learning`

error using the training data samples with the AoI value δ. As the results of the two training methods are similar, the experimental results of the second training method are omitted.

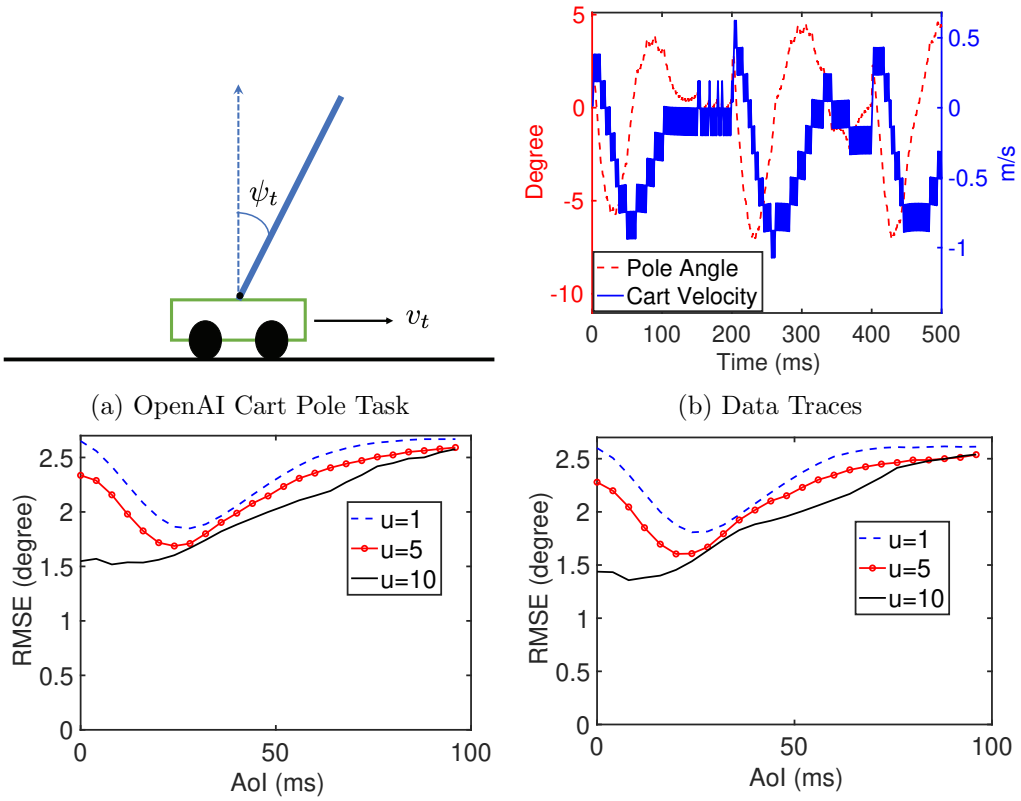
Fig. 2.1 presents the training error and inference error of a video prediction experiment, where a video frame V_t at time t is predicted using a feature $X_{t-\delta} = (V_{t-\delta}, V_{t-\delta-1})$ that is composed of two consecutive video frames. One can observe from Fig. 2.1(b)-(c) that both the training error and the inference error increase as the AoI δ increases.

Fig. 2.2 plots the performance of robot state prediction in a leader-follower robotic system, where a leader robot uses a neural network to predict the follower robot's state Y_t by using the leader robot's state $X_{t-\delta}$ generated δ time slots ago. As depicted in Fig. 2.2, the training and the inference errors decrease in AoI, when AoI ≤ 25 and increase in AoI when AoI ≥ 25. In this case, even a fresh feature with AoI=0 is not good for prediction.

The performance of actuator state prediction under mechanical response delay is depicted in Fig. 2.3. We consider the OpenAI CartPole-v1 task [2], where the objective is to control the force on a cart and prevent the pole attached to the cart from falling over. The pole angle ψ_t at time t is predicted based on a feature $X_{t-\delta} = (v_{t-\delta}, \ldots, v_{t-\delta-u+1})$ that consists of a consecutive sequence of cart velocity with length u generated δ milliseconds (ms) ago. As shown in Fig. 2.3, both the training error and the inference error exhibit non-monotonic variations as the AoI δ increases.

In Fig. 2.4 and Fig. 2.5, we plot the results of temperature prediction and wireless channel state information (CSI) prediction experiments, respectively. In both experiments, we observe non-monotonic trends in training error and inference error with respect to AoI, particularly when the length of the feature sequence u is small.

In the AoI literature, it has been generally assumed that the performance of real-time systems degrades monotonically as the data becomes stale. However, Figs. 2.1-2.5 reveal that this assumption is true in some scenarios, and not true in some other scenarios. Furthermore, Figs. 2.2-2.3 show that even the fresh data with AoI $= 0$ may generate a larger inference error than stale data with AoI > 0. These counter-intuitive experimental results motivated us to seek theoretical interpretations of information freshness in subsequent sections.



(a) OpenAI Cart Pole Task

(b) Data Traces

(c) Training Error vs. AoI

(d) Inference Error vs. AoI

Figure 2.3: Performance of actuator state prediction under mechanical response delay. In the OpenAI CartPole-v1 task [2], the pole angle ψ_t is predicted by using $X_{t-\delta} = (v_{t-\delta}, v_{t-\delta-1}, \ldots, v_{t-\delta-u-1})$, where v_t is the cart velocity at time t and u is the number of past samples or feature sequence length. The training error and inference error are non-monotonic in the AoI.

2.4 Information-theoretic Measures for Remote Inference

In this section, we present information-theoretic measures for evaluating remote inference system.

Because the set of functions Λ constructed by the neural network is complicated, it is difficult to directly analyze the training and inference errors by using (2.1)-(2.3). To overcome this challenge, we introduce information-theoretic metrics for the training and inference errors.

Training Error of the First Training Approach

Let $\Phi = \{f : \mathcal{X} \times \mathbb{Z}^+ \mapsto \mathcal{A}\}$ be the set of all functions mapping from $\mathcal{X} \times \mathbb{Z}^+$ to $\mathcal{A}$. Any action $\phi(x, \delta)$ constructed by the neural network belongs to Φ, whereas the neural network cannot produce some functions in Φ. Hence, $\Lambda \subset \Phi$. By relaxing the set Λ in (2.1) as Φ, we obtain the following lower bound of $\mathrm{err}_{\mathrm{training},1}(\delta)$:

$$\min_{\phi \in \Phi} \mathbb{E}_{Y,X \sim P_{\tilde{Y}_0, \tilde{X}_{-\delta}}}[L(Y, \phi(X, \delta))] = H_L(\tilde{Y}_0 | \tilde{X}_{-\delta}), \tag{2.4}$$

where $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ is a generalized conditional entropy of $\tilde{Y}_0$ given $\tilde{X}_{-\delta}$ [47–49].

Compared to $\mathrm{err}_{\mathrm{training},1}(\delta)$, its information-theoretic lower bound $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ is mathematically more convenient to analyze. The gap between $\mathrm{err}_{\mathrm{training},1}(\delta)$ and $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ was studied recently in [50], where the gap is small if Λ and Φ are close to each other, e.g., when the neural network is sufficiently wide and deep [46].

For notational convenience, we refer to $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ as an *L-conditional entropy*, because it is associated with a loss function L. The *L-entropy* of a random variable Y is defined as [47, 49]

$$H_L(Y) = \min_{a \in \mathcal{A}} \mathbb{E}_{Y \sim P_Y}[L(Y, a)]. \tag{2.5}$$

The L-conditional entropy of Y given $X = x$ is

$$H_L(Y|X = x) = \min_{a \in \mathcal{A}} \mathbb{E}_{Y \sim P_{Y|X=x}}[L(Y, a)]. \tag{2.6}$$

Using (2.4), one can get [47, 49]

$$\begin{aligned} H_L(Y|X) &= \sum_{x\in\mathcal{X}} P_X(x) \min_{a\in\mathcal{A}} \mathbb{E}_{Y\sim P_{Y|X=x}}[L(Y,a)] \\ &= \sum_{x\in\mathcal{X}} P_X(x) H_L(Y|X=x). \end{aligned} \tag{2.7}$$

Training Error of the Second Training Approach

A lower bound of the training error $\text{err}_{\text{training,2}}$ in (2.2) is

$$H_L(\tilde{Y}_0|\tilde{X}_{-\Theta},\Theta) = \min_{\phi\in\Phi} \mathbb{E}_{Y,X,\Theta\sim P_{\tilde{Y}_0,\tilde{X}_{-\Theta},\Theta}}[L(Y,\phi(X,\Theta))], \tag{2.8}$$

where $H_L(\tilde{Y}_0|\tilde{X}_{-\Theta},\Theta)$ is a L-conditional entropy of $\tilde{Y}_0$ given $(\tilde{X}_{-\Theta},\Theta)$. Using (2.7), we get

$$H_L(\tilde{Y}_0|\tilde{X}_{-\Theta},\Theta) = \sum_{x\in\mathcal{X},\delta\in\mathbb{Z}^+} P_{\tilde{X}_{-\Theta},\Theta}(x,\delta) H_L(\tilde{Y}_0|\tilde{X}_{-\delta}=x,\Theta=\delta). \tag{2.9}$$

We assume that the label and feature $(\tilde{Y}_0,\tilde{X}_{-k})$ in the training dataset are independent of the training AoI Θ for every $k\geq 0$. Under this assumption, (2.9) can be simplified as

$$H_L(\tilde{Y}_0|\tilde{X}_{-\Theta},\Theta) = \sum_{\delta\in\mathbb{Z}^+} P_\Theta(\delta)\ H_L(\tilde{Y}_0|\tilde{X}_{-\delta}), \tag{2.10}$$

which is proven in Appendix 2.E.

Inference Error

Let a_{P_Y} be an optimal solution to (2.5), called a *Bayes action* [47]. If the neural predictor in (2.3) is replaced by the Bayes action $a_{P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}}$, then $\text{err}_{\text{inference}}(\delta)$ becomes the following L-conditional cross entropy

$$H_L\left(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}}\right) = \sum_{x\in\mathcal{X}} P_{X_{t-\delta}}(x) \mathbb{E}_{Y\sim P_{Y_t|X_{t-\delta}=x}}\Big[L\left(Y, a_{P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}}\right)\Big], \tag{2.11}$$

where the *L-cross entropy* is defined as

$$H_L(P_Y; P_{\tilde{Y}}) = \mathbb{E}_{Y\sim P_Y}\left[L\left(Y, a_{P_{\tilde{Y}}}\right)\right], \tag{2.12}$$

and the *L-conditional cross entropy* is defined as

$$H_L(P_Y; P_{\tilde{Y}}|P_X) = \sum_{x\in\mathcal{X}} P_X(x)\mathbb{E}_{Y\sim P_{Y|X=x}}\left[L\left(Y, a_{P_{\tilde{Y}|\tilde{X}=x}}\right)\right]. \tag{2.13}$$

If the function spaces Λ and Φ are close to each other, the difference between $\mathrm{err}_{\mathrm{inference}}(\delta)$ and the L-conditional cross entropy $H_L\left(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}}\right)$ is small.

Examples of loss function L, L-entropy, and L-cross entropy are provided in Appendix 2.B. Additionally, the definitions of L-divergence $D_L(P_Y||Q_Y)$, L-mutual information $I_L(Y;X)$, and L-conditional mutual information $I_L(Y;X|Z)$ are discussed below.

The *L-divergence* $D_L(P_Y||P_{\tilde{Y}})$ of P_Y from $P_{\tilde{Y}}$ can be expressed as [47, 49]

$$D_L(P_Y||P_{\tilde{Y}}) = \mathbb{E}_{Y\sim P_Y}\left[L\left(Y, a_{P_Y}\right)\right] - \mathbb{E}_{Y\sim P_Y}\left[L\left(Y, a_{P_{\tilde{Y}}}\right)\right]. \tag{2.14}$$

Because a_{P_Y} is an optimal to (2.5), from (2.41), we have

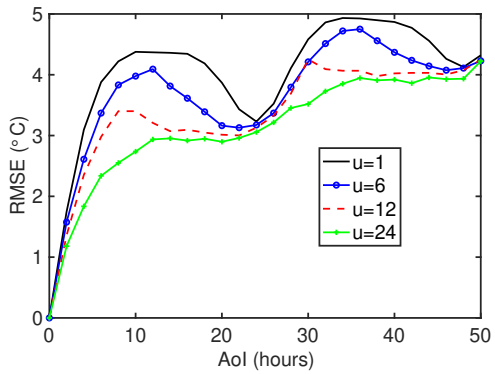
$$D_L(P_Y||P_{\tilde{Y}}) \geq 0. \tag{2.15}$$

The *L-mutual information* $I_L(Y;X)$ is defined as [47, 49]

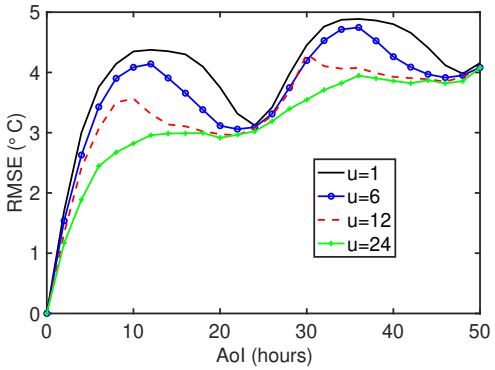
$$\begin{aligned} I_L(Y;X) &= \mathbb{E}_{X\sim P_X}\left[D_L\left(P_{Y|X}||P_Y\right)\right] \\ &= H_L(Y) - H_L(Y|X) \geq 0, \end{aligned} \tag{2.16}$$

which measures the performance gain in predicting Y by observing X. In general, $I_L(Y;X)$ $\neq I_L(X;Y)$. The L-conditional mutual information $I_L(Y;X|Z)$ is given by

$$\begin{aligned} I_L(Y;X|Z) &= \mathbb{E}_{X,Z\sim P_{X,Z}}\left[D_L\left(P_{Y|X,Z}||P_{Y|Z}\right)\right] \\ &= H_L(Y|Z) - H_L(Y|X,Z) \geq 0. \end{aligned} \tag{2.17}$$

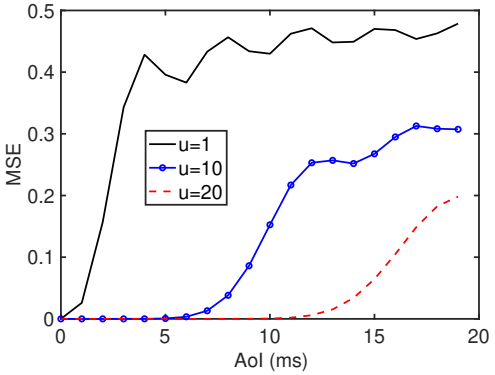


(a) Training Error vs. AoI

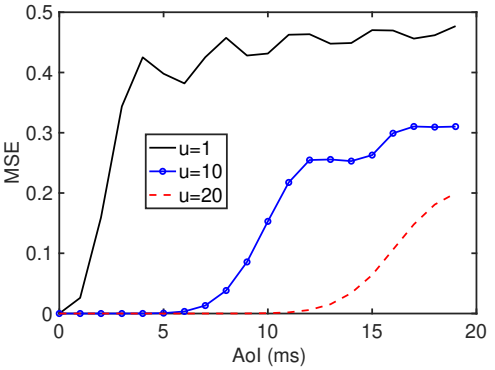


(b) Inference Error vs. AoI

Figure 2.4: Performance of temperature Prediction. The training error and inference error are non-monotonic in AoI. As the feature sequence length u increases, the errors tend closer to non-decreasing functions of the AoI.



(a) Training Error vs. AoI



(b) Inference Error vs. AoI

Figure 2.5: Performance of channel state information (CSI) prediction. The training error and inference error are non-monotonic in AoI. As the feature sequence length u increases, the errors tend closer to non-decreasing functions of the AoI.

In general, $I_L(X;Y) \neq I_L(Y;X)$, which is different from f-mutual information. The relationship among L-divergence, Bregman divergence [51], and f-divergence [52] is discussed in Appendix 2.C.

2.5 An Information-theoretic Interpretation

Training Error vs. Training AoI

We first analyze the monotonocity of L-conditional entropy $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ as δ increases. If $\tilde{Y}_0 \leftrightarrow \tilde{X}_{-\mu} \leftrightarrow \tilde{X}_{-\mu-\nu}$ is a Markov chain for all $\mu, \nu \geq 0$, by the data processing inequality

for L-conditional entropy [48, Lemma 12.1], $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ is a non-decreasing function of δ. Nevertheless, the experimental results in Figs. 2.1-2.5 show that the training error is a growing function of the AoI δ in some systems (see Fig. 2.1), whereas it is a non-monotonic function of δ in other systems (see Figs. 2.2-2.5). As we will explain below, a fundamental reason behind these phenomena is that practical time-series data for remote inference could be either Markovian or non-Markovian. For non-Markovian $(\tilde{Y}_0, \tilde{X}_{-\mu}, \tilde{X}_{-\mu-\nu})$, $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ is not necessarily monotonic in δ.

We propose a new relaxation of the data processing inequality to analyze information freshness for both Markovian and non-Markovian time-series data. To that end, the following relaxation of the standard Markov chain model is needed, which is motivated by the ϵ-dependence concept used in [53].

Definition 2.1 (ϵ-Markov Chain) *Given $\epsilon \geq 0$, a sequence of three random variables Z, X, and Y is said to be an* ϵ-Markov chain*, denoted as $Z \overset{\epsilon}{\to} X \overset{\epsilon}{\to} Y$, if*

$$I_{\log}(Y; Z|X) = D_{\log}\big(P_{Y,X,Z}||P_{Y|X}P_{Z|X}P_X\big) \leq \epsilon^2, \tag{2.18}$$

where[2]

$$D_{\log}(P_Y||Q_Y) = \sum_{y\in\mathcal{Y}} P_Y(y)\log\frac{P_Y(y)}{Q_Y(y)} \tag{2.19}$$

is KL-divergence and $I_{\log}(Y; Z|X)$ is Shannon conditional mutual information.

Notice that the KL-divergence in (2.18) can be also equivalently expressed as

$$\begin{aligned} D_{\log}(P_{Y,X,Z}||P_{Y|X}P_{Z|X}P_X) &= \mathbb{E}_X[D_{\log}(P_{Y,Z|X}||P_{Y|X}P_{Z|X})] \\ &= \mathbb{E}_{X,Z}[D_{\log}(P_{Y|X,Z}||P_{Y|X})], \end{aligned} \tag{2.20}$$

A Markov chain is an ϵ-Markov chain with $\epsilon = 0$. If $Z \to X \to Y$ is a Markov chain, then $Y \to X \to Z$ is also a Markov chain [55, p. 34]. A similar property holds for the ϵ-Markov chain.

[2]In (2.18), if $P_{Y|X=x}(y) = 0$, then $P_{Y|X=x,Z=z}(y) = 0$ which leads to a term $0\log\frac{0}{0}$ in the KL-divergence $D_{\log}(P_{Y|X=x,Z=z}||P_{Y|X=x})$. We adopt the convention in information theory [54] to define $0\log\frac{0}{0} = 0$.

Lemma 2.1 *If $Z \overset{\epsilon}{\to} X \overset{\epsilon}{\to} Y$, then $Y \overset{\epsilon}{\to} X \overset{\epsilon}{\to} Z$.*

Proof 2.1 *See Appendix 2.F.*

By Lemma 2.1, the ϵ-Markov chain can be denoted as $Y \overset{\epsilon}{\leftrightarrow} X \overset{\epsilon}{\leftrightarrow} Z$. In the following lemma, we provide a relaxation of the data processing inequality, which is called an *ϵ-data processing inequality*.

Lemma 2.2 (ϵ-data processing inequality) *The following assertions are true:*

(a) If $Y \overset{\epsilon}{\leftrightarrow} X \overset{\epsilon}{\leftrightarrow} Z$ is an ϵ-Markov chain, then

$$H_L(Y|X) \leq H_L(Y|Z) + O(\epsilon). \tag{2.21}$$

(b) If, in addition, $H_L(Y)$ is twice differentiable in $\mathcal{P}^{\mathcal{Y}}$, then

$$H_L(Y|X) \leq H_L(Y|Z) + O(\epsilon^2). \tag{2.22}$$

Proof 2.2 *Lemma 2.2 is proven by using a local information geometric analysis. See Appendix 2.G for the details.*

Now, we are ready to characterize how $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ varies with the AoI δ.

Theorem 2.1 *The L-conditional entropy*

$$H_L(\tilde{Y}_0|\tilde{X}_{-\delta}) = g_1(\delta) - g_2(\delta) \tag{2.23}$$

is a function of δ, where $g_1(\delta)$ and $g_2(\delta)$ are two non-decreasing functions of δ, given by

$$\begin{aligned} g_1(\delta) =& H_L(\tilde{Y}_0|\tilde{X}_0) + \sum_{k=0}^{\delta-1} I_L(\tilde{Y}_0; \tilde{X}_{-k}|\tilde{X}_{-k-1}), \\ g_2(\delta) =& \sum_{k=0}^{\delta-1} I_L(\tilde{Y}_0; \tilde{X}_{-k-1}|\tilde{X}_{-k}), \end{aligned} \tag{2.24}$$

where the L-conditional mutual information $I_L(Y;X|Z)$ between two random variables Y and X given Z is defined in (2.44). *If $\tilde{Y}_0 \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu} \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu-\nu}$ is an ϵ-Markov chain for every $\mu, \nu \geq 0$, then $g_2(\delta) = O(\epsilon)$ and*

$$H_L(\tilde{Y}_0|\tilde{X}_{-\delta}) = g_1(\delta) + O(\epsilon). \tag{2.25}$$

Proof 2.3 *See Appendix 2.H.*

According to Theorem 2.1, the monotonicity of $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ in δ is characterized by the parameter $\epsilon \geq 0$ in the ϵ-Markov chain model. If ϵ is small, then $\tilde{Y}_0 \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu} \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu-\nu}$ is close to a Markov chain, and $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ is nearly non-decreasing in δ. If ϵ is large, then $\tilde{Y}_0 \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu} \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu-\nu}$ is far from a Markov chain, and $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ could be non-monotonic in δ. Theorem 2.1 can be readily extended to the training error with random AoI Θ by using stochastic orders [56].

Definition 2.2 (Univariate Stochastic Ordering) *[56] A random variable X is said to be stochastically smaller than another random variable Z, denoted as $X \leq_{st} Z$, if*

$$P(X > x) \leq P(Z > x), \quad \forall x \in \mathbb{R}. \tag{2.26}$$

Theorem 2.2 *If $\tilde{Y}_0 \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu} \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu-\nu}$ is an ϵ-Markov chain for all $\mu, \nu \geq 0$, and the training AoIs in two experiments 1 and 2 satisfy $\Theta_1 \leq_{st} \Theta_2$, then*

$$H_L(\tilde{Y}_0|\tilde{X}_{-\Theta_1}, \Theta_1) \leq H_L(\tilde{Y}_0|\tilde{X}_{-\Theta_2}, \Theta_2) + O(\epsilon). \tag{2.27}$$

Proof 2.4 *See Appendix 2.I.*

According to Theorem 2.2, if Θ_1 is stochastically smaller than Θ_2, then the training error in Experiment 1 is approximately smaller than that in Experiment 2. If, in addition to the conditions in Theorems 2.1 and 2.2, $H_L(\tilde{Y}_0)$ is twice differentiable in $P_{\tilde{Y}_0}$, then the last term $O(\epsilon)$ in (2.25) and (2.27) becomes $O(\epsilon^2)$.

Inference Error vs. Inference AoI

Using (2.6), (2.7), and (2.13), it is easy to show that the L-conditional cross entropy $H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}})$ is lower bounded by the L-conditional entropy $H_L(Y_t|X_{t-\delta})$. In addition, $H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}})$ is close to its lower bound $H_L(Y_t|X_{t-\delta})$, if the conditional distributions $P_{Y_t|X_{t-\delta}}$ and $P_{\tilde{Y}_0|\tilde{X}_{-\delta}}$ are close to each other, as stated in Lemma 2.3.

Lemma 2.3 *Given $\beta \geq 0$, if for all $\delta \in \mathbb{Z}^+$*

$$\sum_{x\in\mathcal{X}} P_{X_{t-\delta}}(x) \sum_{y\in\mathcal{Y}} (P_{Y_t|X_{t-\delta}=x}(y) - P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}(y))^2 \leq \beta^2, \tag{2.28}$$

then for all $\delta \in \mathbb{Z}^+$

$$H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}}) = H_L(Y_t|X_{t-\delta}) + O(\beta). \tag{2.29}$$

Proof 2.5 *See Appendix 2.J.*

Combining Theorem 2.1 and Lemma 2.3, the monotonicity of $H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}})$ versus δ is characterized in the next theorem.

Theorem 2.3 *If $Y_t \overset{\epsilon}{\leftrightarrow} X_{t-\mu} \overset{\epsilon}{\leftrightarrow} X_{t-\mu-\nu}$ is an ϵ-Markov chain for all $\mu, \nu \geq 0$ and (2.28) holds for all $\delta \in \mathcal{Z}^+$, then for all $0 \leq \delta_1 \leq \delta_2$*

$$H_L(P_{Y_t|X_{t-\delta_1}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta_1}}|P_{X_{t-\delta_1}}) \leq H_L(P_{Y_t|X_{t-\delta_2}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta_2}}|P_{X_{t-\delta_2}}) + O\big(\max\{\epsilon, \beta\}\big). \tag{2.30}$$

Proof 2.6 *See Appendix 2.K.*

According to Theorem 2.3, $H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}})$ is a function of the AoI δ. If ϵ and β are close to zero, $H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}})$ is nearly a non-decreasing function of δ; otherwise, $H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}_{-\delta}}|P_{X_{t-\delta}})$ can be far from a monotonic function of δ.

Interpretation of the Experimental Results

We use Theorems 2.1-2.3 to interpret the experimental results in Figs. 2.1-2.5. In Fig. 2.1, the training and inference errors for video prediction are increasing functions of the AoI. This observation suggests that the target and feature time-series data $(Y_t, X_{t-\mu}, X_{t-\mu-\nu})$ for

video prediction is close to a Markov chain. In the robot state prediction experiment depicted in Fig. 2.2, the state of the follower robot depends on the state of the leader robot received through a channel. Due to the communication delay from the leader robot to the follower robot, the target and feature data sequence $(Y_t, X_{t-\mu}, X_{t-\mu-\nu})$ can be far from a Markov chain. In the experiment of actuator state prediction under mechanical response delay, pole angle at time t is strongly correlated with the cart velocity generated 25 ms ago, as observed from data traces in Fig. 2.3(b). Moreover, temperature and CSI signals have long-range dependence. For example, temperature at time t depends on the temperature of 24 hours ago. These observations imply that the target and feature data sequence $(Y_t, X_{t-\mu}, X_{t-\mu-\nu})$ for all $\mu, \nu \geq 0$ may not be close to a Markov chain in the experimental results depicted in Figs. 2.2-2.5. Because the target and feature time-series data involved is non-Markovian, Theorems 2.1-2.3 suggest that the training error and inference error could be non-monotonic with respect to AoI, as observed in Figs. 2.2-2.5.

Recall that u is the sequence length of the feature $X_t = (V_t, V_{t-1}, \ldots, V_{t-u+1})$. In Figs. 2.3-2.5, the training and inference errors tends to become non-decreasing functions of the AoI δ as the feature length u grows. This phenomenon can be interpreted by Theorems 2.1-2.3: According to Shannon's Markovian representation of discrete-time sources in his seminal work [57], the larger u, the closer $(Y_t, X_{t-\mu}, X_{t-\mu-\nu})$ tends to a Markov chain. According to Theorems 2.1-2.3, as u increases, the training and inference errors tends to be non-decreasing with respect to the AoI δ, which agrees with Figs. 2.3-2.5. One disadvantage of large feature length u is that it increases the channel resources needed for transmitting the features. The optimal choice of the feature length u is studied in [34].

2.6 A Model-based Interpretation

We construct two models to analyze the non-monotonocity of the L-conditional entropy $H_L(Y_t|X_{t-\delta})$ with respect to the AoI δ and interpret the reasons.

2.6.1 Reaction Prediction with Delay

To facilitate understanding of the counter-intuitive experimental results illustrated in Figs. 2.2-2.3, we present the following analytical example for reaction prediction.

Example 2.1 (Reaction Prediction) *Consider a causal system represented by* $Y_t = f(X_{t-d})$, *where* X_t *and* Y_t *are the input and output of the system, respectively,* $d \geq 0$ *is the delay introduced by the system, and* $f(\cdot)$ *is a function.*

Lemma 2.4 *If* X_t *is a Markov chain and* $Y_t = f(X_{t-d})$, *then* $H_L(Y_t|X_{t-\delta})$ *decreases with* δ *when* $0 \leq \delta \leq d$ *and increases with* δ *when* $\delta \geq d$. *In addition, for any random variable* Z,

$$H_L(Y_t|X_{t-d}) = H_L(Y_t|Y_t) \leq H_L(Y_t|Z). \tag{2.31}$$

Proof 2.7 *See Appendix 2.M.*

Lemma 2.4 implies that the feature X_{t-d} achieves the minimum expected loss in predicting Y_t. Therefore, for predicting Y_t, X_{t-d} is the optimal choice, not the freshest feature X_t. Hence, fresh data is not always the best.

The robotic state prediction and the actuator state prediction experiments in Figs. 2.2-2.3 are also instances of reaction prediction. Similar to Example 2.1, the freshest feature with AoI=0 is not the best choice for predicting the reaction in Figs. 2.2-2.3. However, the relationship between the leader and follower robots' states in the robotic state prediction experiment and the relationship between cart velocity and pole angle in the actuator state prediction experiment are much more complicated than the input-output relationship in Example 2.1.

2.6.2 Autoregressive Model

We consider that the target Y_t evolves as

$$Y_t = V_t + N_t, \tag{2.32}$$

where $V_t \in \mathbb{R}$ follows a discrete-time p-th order autoregressive (AR(p)) linear time-invariant system:

$$V_t = a_1 V_{t-1} + a_2 V_{t-2} + \ldots + a_p V_{t-p} + W_t, \tag{2.33}$$

$N_t \in \mathbb{R}$ and $W_t \in \mathbb{R}$ are i.i.d. Gaussian noises over time with zero mean, and $a_k \in \mathbb{R}$ for all $k = 1, 2, \ldots, p$. Let $\sigma^2_{Y_t}$ and $\sigma^2_{V_t}$ be the variances of Y_t and V_t, respectively. Our goal is to predict Y_t based on $X_{t-\delta} = (V_{t-\delta}, V_{t-\delta-1}, \ldots, V_{t-\delta-u+1})$, where u is the feature length. For the simplicity of analysis, we use vector $\mathbf{V}_t^u$ instead of X_t in this section, where

$$X_t = \mathbf{V}_t^u = (V_t, V_{t-1}, \ldots, V_{t-u+1}) \tag{2.34}$$

and

$$X_{t-\delta} = \mathbf{V}_{t-\delta}^u = (V_{t-\delta}, V_{t-\delta-1}, \ldots, V_{t-\delta-u+1}). \tag{2.35}$$

We will evaluate the L-conditional entropy associated with two loss functions: (i) Logarithmic Loss (log loss) and (ii) Quadratic loss.

Logarithmic Loss (log loss)

For log loss $L_{\log}(y, Q_{Y_t}) = -\log Q_{Y_t}(y)$, the L-entropy of a continuous random variable Y is the differential entropy [54, 55], defined as

$$H_{\log}(Y_t) = -\int_{y \in \mathbb{R}} p_{Y_t}(y) \log p_{Y_t}(y) \, \mathrm{d}\, y, \tag{2.36}$$

where p_{Y_t} is the density function of the distribution P_{Y_t} of Y_t. Because Y_t is a Gaussian random variable with zero mean, one can obtain [54]

$$H_{\log}(Y_t) = \frac{1}{2} \log \left(2\pi e \, \mathbb{E}[Y_t^2] \right). \tag{2.37}$$

The L-entropy for a discrete random variable associated with log loss is the well known Shannon entropy [12, 14, 49]. The Shannon entropy is always non-negative. However, the differential entropy can be negative, positive, or zero [54].

Proposition 2.1 *The L-conditional entropy $H_{\log}(Y_t|\mathbf{V}_{t-\delta}^u)$ is given by*

$$H_{\log}(Y_t|\mathbf{V}_{t-\delta}^u) = \frac{1}{2}\log\left(\frac{\det(\mathbf{R}_{[Y_t,\mathbf{V}_{t-\delta}^u]})}{\det(\mathbf{R}_{\mathbf{V}_t^u})}\right) + \frac{1}{2}\log 2\pi e, \tag{2.38}$$

where $\det(\mathbf{A})$ *denotes the determinant of a square matrix* $\mathbf{A}$*,*

$$\mathbf{R}_{\mathbf{V}_t^u} = \mathbb{E}[(\mathbf{V}_t^u)^T\mathbf{V}_t^u] \tag{2.39}$$

is an $u \times u$ dimensional auto-correlation matrix of a random vector $\mathbf{V}_t^u$, and

$$\mathbf{R}_{[Y_t,\mathbf{V}_{t-\delta}^u]} = \mathbb{E}\left[[Y_t, \mathbf{V}_{t-\delta}^u]^T[Y_t, \mathbf{V}_{t-\delta}^u]\right] \tag{2.40}$$

is an $(u+1) \times (u+1)$ dimensional auto-correlation matrix of a random vector $[Y_t, \mathbf{V}_{t-\delta}^u] = [Y_t, V_{t-\delta}, \ldots, V_{t-\delta-u+1}]$.

Proof 2.8 *We begin with the definitions of* L-divergence *and* L-mutual information*. The L-divergence $D_L(P_Y||Q_Y)$ of P_Y from Q_Y can be expressed as [12, 47, 49]*

$$D_L(P_Y||Q_Y) = \mathbb{E}_{Y\sim P_Y}\left[L\left(Y, a_{P_Y}\right)\right] - \mathbb{E}_{Y\sim P_Y}\left[L\left(Y, a_{Q_Y}\right)\right], \tag{2.41}$$

where a_{P_Y} is the optimal solution to

$$\min_{a\in\mathcal{A}} \mathbb{E}_{Y\sim P_Y}[L(Y, a)]. \tag{2.42}$$

The L-mutual information *$I_L(Y;X)$ is defined as [12, 47, 49]*

$$\begin{aligned} I_L(Y;X) =& \mathbb{E}_{X\sim P_X}\left[D_L\left(P_{Y|X}||P_Y\right)\right] \\ =& H_L(Y) - H_L(Y|X) \geq 0, \end{aligned} \tag{2.43}$$

which measures the performance gain in estimating Y by observing X. The L-conditional mutual information $I_L(Y;X|Z)$ is given by

$$\begin{aligned} I_L(Y;X|Z) =& \mathbb{E}_{X,Z\sim P_{X,Z}}\left[D_L\left(P_{Y|X,Z}||P_{Y|Z}\right)\right] \\ =& H_L(Y|Z) - H_L(Y|X,Z) \geq 0. \end{aligned} \tag{2.44}$$

Using (2.43)*, the L-conditional entropy $H_{\log}(Y_t|\mathbf{V}_{t-\delta}^u)$ associated with log loss can be expressed as*

$$H_{\log}(Y_t|\mathbf{V}_{t-\delta}^u) = H_{\log}(Y_t) - I_{\log}(Y_t;\mathbf{V}_{t-\delta}^u). \tag{2.45}$$

For jointly Gaussian random vectors $\mathbf{Y}\in\mathbb{R}^m$ and $\mathbf{X}\in\mathbb{R}^n$, we can obtain [54]

$$I_{\log}(\mathbf{Y};\mathbf{X}) = \frac{1}{2}\log\frac{\det(\Sigma_{\mathbf{X}})\det(\Sigma_{\mathbf{Y}})}{\det(\Sigma_{[\mathbf{X},\mathbf{Y}]})}, \tag{2.46}$$

where $\Sigma_{\mathbf{X}} := \mathbb{E}[(\mathbf{X}-\mathbb{E}[\mathbf{X}])]^T\mathbb{E}[(\mathbf{X}-\mathbb{E}[\mathbf{X}])]$ denotes the covariance matrix of the row vector $\mathbf{X}$. If $\mathbb{E}[\mathbf{X}]=0$, then $\Sigma_{\mathbf{X}}=\mathbf{R}_{\mathbf{X}}$. By using $\mathbb{E}[Y_t]=0$, $\mathbb{E}[\mathbf{V}_{t-\delta}^u]=0$, (2.37)*,* (2.45)*, and* (2.46)*, we obtain* (2.38)*. This completes the proof.* ■

In the special case of feature length $u=1$, from (2.38), it can be shown that

$$H_{\log}(Y_t|V_{t-\delta}) = \frac{1}{2}\left(\log\big(\mathbb{E}[Y_t^2] - \frac{\mathbb{E}[V_tV_{t-\delta}]^2}{\mathbb{E}[V_t^2]}\big) + \log 2\pi e\right). \tag{2.47}$$

Quadratic Loss

For quadratic loss function $L_2(y,\hat{y}) = (y-\hat{y})^2$, the L-entropy of Y_t is the variance of Y_t, given by

$$H_2(Y_t) = \sigma_{Y_t}^2. \tag{2.48}$$

Because $\mathbb{E}[Y_t]=0$, we have

$$H_2(Y_t) = \mathbb{E}[Y_t^2]. \tag{2.49}$$

Proposition 2.2 *The L-conditional entropy $H_2(Y_t|\mathbf{V}_{t-\delta}^u)$ is given by*

$$\begin{aligned} H_2(Y_t|\mathbf{V}_{t-\delta}^u) &= \mathbb{E}[(Y_t - \mathbb{E}[Y_t|\mathbf{V}_{t-\delta}^u])^2] \\ &= \mathbb{E}[Y_t^2] - \mathbb{E}[X_t\mathbf{V}_{t-\delta}^u](\mathbf{R}_{\mathbf{V}_t^u})^{-1}\mathbb{E}[X_t\mathbf{V}_{t-\delta}^u]^T, \end{aligned} \tag{2.50}$$

where $\mathbb{E}[V_t\mathbf{V}_{t-\delta}^u] = \big[\mathbb{E}[V_tV_{t-\delta}], \ldots, \mathbb{E}[V_tV_{t-\delta-u+1}]\big]$ is a $1 \times u$ dimensional vector and $\mathbf{R}_{\mathbf{V}_t^u}$ is an $u \times u$ dimensional auto-correlation matrix of $\mathbf{V}_t^u$ defined in (2.39)*.*

Proof 2.9 *The conditional expectation of Y_t given $\mathbf{V}_{t-\delta}^u = \mathbf{v}^u$, i.e., $\mathbb{E}[Y_t|\mathbf{V}_{t-\delta}^u = \mathbf{v}^u]$ is the optimal estimator of*

$$\min_{\phi(\mathbf{v}^u,\delta)\in\mathbb{R}} \mathbb{E}_{Y\sim P_{Y_t|\mathbf{v}_{t-\delta}^u=v^u}}\Big[(Y, \phi(\mathbf{v}^u, \delta))^2\Big]. \tag{2.51}$$

By substituting $L(y, \phi(\mathbf{v}^u, \delta)) = (y - \phi(\mathbf{v}^u, \delta))^2$ and $\phi(\mathbf{v}^u, \delta) = \mathbb{E}[Y_t|\mathbf{V}_{t-\delta}^u = \mathbf{v}^u]$ into (**??**)*, we obtain*

$$H_2(Y_t|\mathbf{V}_{t-\delta}^u) = \mathbb{E}[(Y_t - \mathbb{E}[Y_t|\mathbf{V}_{t-\delta}^u])^2]. \tag{2.52}$$

Since Y_t and $\mathbf{V}_{t-\delta}^u$ are jointly Gaussian with $\mathbb{E}[Y_t] = 0$, $\mathbb{E}[\mathbf{V}_{t-\delta}^u] = 0$, and

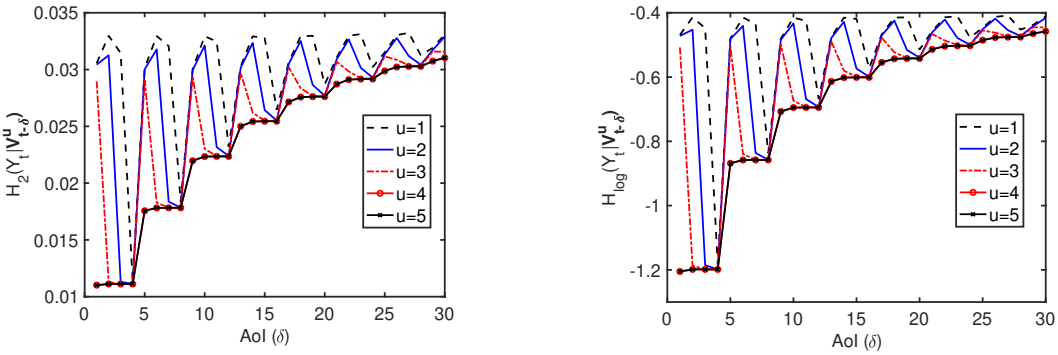
$$\mathbb{E}[Y_tV_{t-k}] = \mathbb{E}[V_tV_{t-k}] + \mathbb{E}[N_tV_{t-k}] = \mathbb{E}[V_tV_{t-k}], \tag{2.53}$$

we get [58, Chapter 7.3]

$$\mathbb{E}[Y_t|\mathbf{V}_{t-\delta}^u = \mathbf{v}^u] = \mathbf{A}(\mathbf{v}^u)^T, \tag{2.54}$$

where

$$\mathbf{A} = \mathbb{E}[V_t\mathbf{V}_{t-\delta}^u](\mathbf{R}_{\mathbf{V}_t^u})^{-1}. \tag{2.55}$$



(a) $H_2(Y_t|\mathbf{V}^u_{t-\delta})$ vs. AoI (δ) (b) $H_{\log}(Y_t|\mathbf{V}^u_{t-\delta})$ vs. AoI (δ)

Figure 2.6: L-conditional entropy vs. AoI with (a) quadratic loss function and (b) log loss function (base 2). The L-conditional entropy is not always a monotonic function of AoI. An AR(4) model as defined in (2.67)-(2.68) is considered for this simulation.

By using orthogonality principle [58, Chapter 7.3], we get

$$\mathbb{E}[(Y_t - \mathbf{A}(\mathbf{V}^u_{t-\delta})^T)\mathbf{V}^u_{t-\delta}] = 0 \tag{2.56}$$

Now, using (2.54) *and* (2.56)*, we obtain from* (2.52) *that*

$$\begin{aligned} H_2(Y_t|\mathbf{V}^u_{t-\delta}) &= \mathbb{E}[(Y_t - \mathbb{E}[Y_t|\mathbf{V}^u_{t-\delta}])^2] \\ &= \mathbb{E}[(Y_t - \mathbf{A}(\mathbf{V}^u_{t-\delta})^T)Y_t] \\ &= \mathbb{E}[Y_t^2] - \mathbf{A}(\mathbb{E}[Y_t\mathbf{V}^u_{t-\delta}])^T \\ &= \mathbb{E}[Y_t^2] - \mathbf{A}(\mathbb{E}[V_t\mathbf{V}^u_{t-\delta}])^T. \end{aligned} \tag{2.57}$$

By substituting (2.55) *into* (2.57)*, we obtain* (2.50)*. This completes the proof.* ■

In the special case of feature length $u = 1$, from (2.50), it can be shown that

$$H_2(Y_t|X_{t-\delta}) = \mathbb{E}[Y_t^2] - \frac{\mathbb{E}[X_t\mathbb{X}^u_{t-\delta}]^2}{\mathbb{E}[X_t^2]}. \tag{2.58}$$

By utilizing Propositions 2.1-2.2, one can evaluate the L-conditional entropy of a data sequence that is generated using a Gaussian AR(p) system.

Characterizing the Parameter ϵ of An ϵ-Markov Chain

In this section, we show how to evaluate the value of the parameter ϵ from an AR(p) process. We also analyzed the impact of feature length u on the parameter ϵ.

The parameter ϵ in $Y_t \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu}^u \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu-\nu}^u$ depends on μ, ν, and u. We denote $\epsilon_{\mu,\nu}(u)$ as the minimum value of ϵ for which $Y_t \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu}^u \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu-\nu}^u$ is an ϵ-Markov chain. By using Definition 2.1, we have

$$\epsilon_{\mu,\nu}(u) = \sqrt{I_{\log}(Y_t; \mathbf{V}_{t-\mu-\nu}^u | \mathbf{V}_{t-\mu}^u)}. \tag{2.59}$$

We also denote $\epsilon(u)$ as the minimum value of ϵ for which $Y_t \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu}^u \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu-\nu}^u$ is an ϵ-Markov chain for all $\mu, \nu \geq 0$. Then, we can write

$$\epsilon(u) = \max_{\mu,\nu \geq 0} \epsilon_{\mu,\nu}(u). \tag{2.60}$$

Proposition 2.3 *The following assertions are true for the Gaussian AR(p) model defined in* (2.32)-(2.33)*.*

(a) The minimum value of ϵ for which the data sequence $(Y_t, \mathbf{V}_{t-\mu}^u, \mathbf{V}_{t-\mu-\nu}^u)$ satisfies an ϵ-Markov chain property, i.e., $Y_t \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu}^u \overset{\epsilon}{\leftrightarrow} \mathbf{V}_{t-\mu-\nu}^u$, for all $\mu, \nu \geq 0$ is given by

$$\epsilon(u) = \max_{\mu,\nu \geq 0} \epsilon_{\mu,\nu}(u), \tag{2.61}$$

where $\epsilon_{\mu,\nu}(u)$ is determined by

$$\epsilon_{\mu,\nu}(u) = \sqrt{\frac{1}{2} \log \left(\frac{\det(\mathbf{R}_{[\mathbf{V}_{t-\mu-\nu}^u, \mathbf{V}_{t-\mu}^u]}) \det(\mathbf{R}_{[Y_t, \mathbf{V}_{t-\mu}^u]})}{\det(\mathbf{R}_{\mathbf{V}_{t-\mu}^u}) \det(\mathbf{R}_{[Y_t, \mathbf{V}_{t-\mu-\nu}^u, \mathbf{V}_{t-\mu}^u]})} \right)}, \tag{2.62}$$

$\det(\mathbf{A})$ *denotes the determinant of a square matrix $\mathbf{A}$, and $\mathbf{R}_{\mathbf{X}} = \mathbb{E}[\mathbf{X}^T \mathbf{X}]$ is the auto-correlation matrix of a random vector $\mathbf{X}$.*

(b) If $u \geq p$, then

$$\epsilon(u) = 0. \tag{2.63}$$

Proof 2.10 *Part(a): The Shannon's conditional mutual information* $I_{\log}(Y_t; \mathbf{V}^u_{t-\mu-\nu}|\mathbf{V}^u_{t-\mu})$ *can be derived as follows:*

$$\begin{aligned}
I_{\log}(Y_t; \mathbf{V}^u_{t-\mu-\nu}|\mathbf{V}^u_{t-\mu}) \overset{(a)}{=} & H_{\log}(Y_t|\mathbf{V}^u_{t-\mu}) - H_{\log}(Y_t|\mathbf{V}^u_{t-\mu}, \mathbf{V}^u_{t-\mu-\nu}) \\
= & H_{\log}(Y_t) - H_{\log}(Y_t|\mathbf{V}^u_{t-\mu}, \mathbf{V}^u_{t-\mu-\nu}) \\
& - H_{\log}(Y_t) + H_{\log}(Y_t|\mathbf{V}^u_{t-\mu}) \\
\overset{(b)}{=} & I_{\log}(Y_t; \mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu}) - I_{\log}(Y_t; \mathbf{V}^u_{t-\mu}) \\
\overset{(c)}{=} & \frac{1}{2}\log\left(\frac{\det(\Sigma_{[\mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu}]})\det(\Sigma_{Y_t})}{\det(\Sigma_{[Y_t, \mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu})]}}\right) \\
& - \frac{1}{2}\log\left(\frac{\det(\Sigma_{\mathbf{V}^u_{t-\mu}})\det(\Sigma_{Y_t})}{\det(\Sigma_{[Y_t, \mathbf{V}^u_{t-\mu}]})}\right) \\
= & \frac{1}{2}\log\left(\frac{\det(\Sigma_{[\mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu}]})\det(\Sigma_{[Y_t, \mathbf{V}^u_{t-\mu}]})}{\det(\Sigma_{\mathbf{V}^u_{t-\mu}})\det(\Sigma_{[Y_t, \mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu}]})}\right) \\
\overset{(d)}{=} & \frac{1}{2}\log\left(\frac{\det(\mathbf{R}_{[\mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu}]})\det(\mathbf{R}_{[Y_t, \mathbf{V}^u_{t-\mu}]})}{\det(\mathbf{R}_{\mathbf{V}^u_{t-\mu}})\det(\mathbf{R}_{[Y_t, \mathbf{V}^u_{t-\mu-\nu}, \mathbf{V}^u_{t-\mu}]})}\right), \qquad (2.64)
\end{aligned}$$

where (a), (b), and (c) hold due to (2.44), (2.43), *and* (2.46), *respectively and (d) holds because of* $\mathbb{E}[Y_t] = 0$ *and* $\mathbb{E}[V_{t-k}] = 0$ *for all* k.

Now, by substituting (2.64) *into* (2.59), *we get* (2.62).

Part (b): If $l \geq p$, *by using* (2.33), *we can express* $\mathbf{V}^u_{t-\mu}$ *as a function of* $\mathbf{V}^u_{t-\mu-\nu}$ *for any* $\mu, \nu \geq 0$. *Hence, if* $l \geq p$, *then*

$$H_L(Y_t|\mathbf{V}^u_{t-\mu}, \mathbf{V}^u_{t-\mu-\nu}) = H_L(Y_t|\mathbf{V}^u_{t-\mu}). \qquad (2.65)$$

By using (2.44) *and* (2.65), *we get that if* $l \geq p$, *then*

$$I_L(Y_t; \mathbf{V}^u_{t-\mu-\nu}|\mathbf{V}^u_{t-\mu}) = 0. \qquad (2.66)$$

From (2.62) *and* (2.66), *we obtain that if* $l \geq p$, *then* $\epsilon_{\mu,\nu}(l) = 0$ *for all* $\mu, \nu \geq 0$. *Thus,* $\epsilon(l) = 0$. *This concludes the proof.* ■

In Proposition 2.3(a), we present a closed-form expression for computing the parameter $\epsilon(u)$. Utilizing Proposition 2.3(a), one can derive $\epsilon(u)$ from the auto-correlation function of a data sequence generated from the AR(p) model. Proposition 2.3(b) implies that if the feature length u is greater than or equal to the order p of the AR(p) model, then $\epsilon(u)$ equals 0. By integrating Proposition 2.3(b) with Theorem 2.1, we can conclude that if the feature length u is greater than or equal to the order p, the L-conditional entropy becomes a non-decreasing function of AoI. However, transmitting longer features demands more communication resources [34].

We evaluate the L-conditional entropy using the following autoregressive linear system model.

Example 2.2 (Autoregressive Model) *We utilize Proposition 2.1-2.3 to compute the L-conditional entropy and the parameter $\epsilon(u)$ for the following AR(4) process:*

$$V_t = 0.1V_{t-1} + 0.8V_{t-p} + W_t, \tag{2.67}$$

$$Y_t = V_t + N_t, \tag{2.68}$$

where $W_t \in \mathbb{R}$ and $N_t \in \mathbb{R}$ are i.i.d. Gaussian noises over time with zero mean and variances 0.01 and 0.001, respectively. The goal is to estimate Y_t using a feature sequence $\mathbf{V}_{t-\delta}^{u} = [V_{t-\delta}, V_{t-\delta-1}, \ldots, V_{t-\delta-u+1}]$.

We compute the L-conditional entropy of Y_t given $\mathbf{V}_{t-\delta}^{u}$ for two different loss functions: (a) quadratic loss and (b) log loss, using (2.50) and (2.38), respectively. In Fig. 2.6, we illustrate the L-conditional entropy $H_2(Y_t|\mathbf{V}_{t-\delta}^{u})$ associated with quadratic loss and the L-conditional entropy $H_{\log}(Y_t|\mathbf{V}_{t-\delta}^{u})$ associated with log loss (base 2). Both $H_2(Y_t|\mathbf{V}_{t-\delta}^{u})$ and $H_{\log}(Y_t|\mathbf{V}_{t-\delta}^{u})$ exhibit similar behavior with respect to AoI δ and feature length l, but they are measured in different scale and are used in different applications.

Moreover, we determine $\epsilon(u)$ through the following steps: Firstly, we calculate $\epsilon_{\mu,\nu}(u)$ using (2.62) given μ, ν, and u. Subsequently, we compute $\epsilon(u)$ by maximizing $\epsilon_{\mu,\nu}(u)$ over all $\mu, \nu \geq 0$. However, this needs to compute $\epsilon_{\mu,\nu}(u)$ for an infinite number of μ and ν, which is not possible. We find that when μ or ν exceed a large value, $\epsilon_{\mu,\nu}(u)$ becomes either 0 or close to 0 for all l. Therefore, we can choose an upper bound denoted as M and compute

$\epsilon(u)$ by maximizing $\epsilon_{\mu,\nu}(u)$ over all $0 \leq \mu, \nu \leq M$. In our simulation, we set $M = 50$. The outcomes of $\epsilon(l)$ for feature length $l = 1, 2, 3, 4, 5$ are 1.55, 1.49, 1.39, 0, and 0, respectively.

Fig. 2.6 and the value of $\epsilon(u)$ illustrate that as the feature length u increases, the parameter $\epsilon(u)$ tends to zero, and the L-conditional entropy becomes a monotonic function of AoI δ. Specifically, when the feature length u reaches the order p of the AR(p) process, the parameter $\epsilon(u)$ equals zero and hence, the L-conditional entropy becomes a monotonic function of AoI. Moreover, as the feature length u increases, the L-conditional entropy reduces. However, beyond the order p, further increases in feature length do not result in the reduction of the L-conditional entropy. It is evident from Fig. 2.6 that the L-conditional entropy for $u = 4$ and $u = 5$ remains the same for the AR(4) model.

2.7 Conclusions

In this chapter, we explored the impact of data freshness on the performance of remote inference systems. We conducted experimental studies and provided information-theoretical analysis to reveal that the inference error in a remote inference system is a function of AoI, but not necessarily a monotonic function of AoI. If the target and feature data sequence satisfy a Markov chain, then the inference error is a monotonic function of AoI. Otherwise, if the target and feature data sequence is far from a Markov chain, then the inference error can be a non-monotonic function of AoI. We also verified our results by constructing two analytical models.

Appendix

2.A Experimental Setup for ML Experiments in Chapter 2.3

In all five experiments, we employed the first training method described in Section 2.2. This approach involves training multiple neural networks independently and in parallel, each using a distinct dataset with a different AoI value. In contrast, the second approach trains a single neural network on a larger, combined dataset encompassing various AoI values. Due to the smaller dataset sizes for each network, the first approach can potentially have a shorter training time than the second approach. The experimental settings of the five experiments are provided below:

Video Prediction: In video prediction experiment, a pre-trained neural network model called "SAVP" [1] is used to evaluate on 256 samples of "BAIR" dataset [59], which contains video frames of a randomly moving robotic arm. The pre-trained neural network model can be downloaded from the GitHub repository of [1].

Robot State Prediction: In this experiment, we consider a leader-follower robotic system illustrated in a YouTube video [3], where we used two Kinova JACO robotic arms with 7 degrees of freedom and 3 fingers to accomplish a pick and place task. The leader robot sends its state (7 joint angles and positions of 3 fingers) X_t to the follower robot through a channel. One packet for updating the leader robot's state is sent periodically to the follower robot every 20 time-slots. The transmission time of each updating packet is 20 time-slots. The follower robot moves towards the leader's most recent state and locally controls its robotic fingers to grab an object. We constructed a robot simulation environment using the Robotics System Toolbox in MATLAB. In each episode, a can is randomly generated on a table in front of the follower robot. The leader robot observes the position of the can and illustrates to the follower robot how to grab the can and place it on another table, without colliding with other objects in the environment. The rapidly-exploring random tree (RRT) algorithm is used to control the leader robot [60]. For the local control of the follower robot, an interpolation method is used to generate a trajectory between two points sent from the leader robot while also avoiding collisions with other obstacles. The leader robot uses a neural network to predict the follower robot's state Y_t. The neural network consists of

[3] `https://youtu.be/_z4FHuu3-ag`

one input layer, one hidden layer with 256 ReLU activation nodes, and one fully connected (dense) output layer. The dataset contains the leader and follower robots' states in 300 episodes of continue operation. The first 80% of the dataset is used for the training and the other 20% of the dataset is used for the inference.

Actuator State Prediction: We consider the OpenAI CartPole-v1 task [2], where a DQN reinforcement learning algorithm [61] is used to control the force on a cart and keep the pole attached to the cart from falling over. By simulating 10^4 episodes of the OpenAI CartPole-v1 environment, a time-series dataset is collected that contains the pole angle ψ_t and the velocity v_t of the cart. The pole angle ψ_t at time t is predicted based on a feature $X_{t-\delta} = (v_{t-\delta}, \ldots, v_{t-\delta-u+1})$, i.e., a vector of cart velocity with length u, where v_t is the cart velocity at time t and $\Delta(t) = \delta$ is the AoI. The predictor in this experiment is an LSTM neural network that consists of one input layer, one hidden layer with 64 LSTM cells, and a fully connected output layer. First 72% of the dataset is used for training and the rest of the dataset is used for inference.

Temperature prediction: The temperature Y_t at time t is predicted based on a feature $X_{t-\delta} = \{s_{t-\delta}, \ldots, s_{t-\delta-u+1}\}$, where s_t is a 7-dimensional vector consisting of temperature, pressure, saturation vapor pressure, vapor pressure deficit, specific humidity, airtightness, and wind speed at time t. We used the Jena climate dataset recorded by the Max Planck Institute for Biogeochemistry [62]. The dataset comprises 14 features, including temperature, pressure, humidity, etc., recorded once every 10 minutes from 10 January 2009 to 31 December 2016. The first 75% of the dataset is used for training and the later 25% is used for inference. Temperature is predicted every hour using an LSTM neural network composed of one input layer, one hidden layer with 32 LSTM units, and one output layer.

CSI Prediction: The CSI h_t at time t is predicted based on a feature $X_{t-\delta} = \{h_{t-\delta}, \ldots, h_{t-\delta-u+1}\}$. The dataset for CSI is generated by using Jakes model [63].

2.B Examples of Loss function L, L-entropy, and L-cross entropy

Several examples of loss function L, L-entropy, and L-cross entropy are listed below. Additional examples can be found in [47–49].

Logarithmic Loss (log-loss)

The log-loss function is given by $L_{\log}(y, Q_Y) = -\log Q_Y(y)$, where the action $a = Q_Y$ is a distribution in $\mathcal{P}^{\mathcal{Y}}$. The corresponding L-entropy is the well-known Shannon's entropy [55], defined as

$$H_{\log}(Y) = -\sum_{y\in\mathcal{Y}} P_Y(y)\ \log\ P_Y(y), \tag{2.69}$$

where P_Y is the distribution of Y. The corresponding L-cross entropy is given by

$$H_{\log}(Y;\tilde{Y}) = -\sum_{y\in\mathcal{Y}} P_Y(y)\ \log\ P_{\tilde{Y}}(y). \tag{2.70}$$

The L-mutual information and L-divergence associated with the log-loss are Shannon's mutual information and the K-L divergence defined in (2.19), respectively.

Brier Loss

The Brier loss function is defined as $L_B(y, Q_Y) = \sum_{y'\in\mathcal{Y}} Q_Y(y')^2 - 2\ Q_Y(y) + 1$ [47]. The associated L-entropy is given by

$$H_B(Y) = 1 - \sum_{y\in\mathcal{Y}} P_Y(y)^2, \tag{2.71}$$

and the associated L-cross entropy is

$$H_B(Y;\tilde{Y}) = \sum_{y\in\mathcal{Y}} P_{\tilde{Y}}(y)^2 - 2\sum_{y\in\mathcal{Y}} P_{\tilde{Y}}(y)P_Y(y) + 1. \tag{2.72}$$

0-1 Loss

The 0-1 loss function is given by $L_{0\text{-}1}(y,\hat{y}) = \mathbf{1}(y \neq \hat{y})$, where $\mathbf{1}(A)$ is the indicator function of event A. For this case, we have

$$H_{0\text{-}1}(Y) = 1 - \max_{y\in\mathcal{Y}} P_Y(y), \tag{2.73}$$

$$H_{0\text{-}1}(Y;\tilde{Y}) = 1 - P_Y\left(\arg\max_{y\in\mathcal{Y}} P_{\tilde{Y}}(y)\right). \tag{2.74}$$

α-Loss

The α-loss function is defined by $L_\alpha(y,Q_Y) = \frac{\alpha}{\alpha-1}\left[1 - Q_Y(y)^{\frac{\alpha-1}{\alpha}}\right]$ for $\alpha > 0$ and $\alpha \neq 1$ [64, Eq. 14]. It becomes the log-loss function in the limit $\alpha \to 1$ and the 0-1 loss function in the limit $\alpha \to \infty$. The L-entropy and L-cross entropy associated with the α-loss function are given by

$$H_{\alpha\text{-loss}}(Y) = \frac{\alpha}{\alpha-1}\left[1 - \left(\sum_{y\in\mathcal{Y}} P_Y(y)^\alpha\right)^{\frac{1}{\alpha}}\right], \tag{2.75}$$

$$H_{\alpha\text{-loss}}(Y;\tilde{Y}) = \frac{\alpha}{\alpha-1}\left[1 - \left(\sum_{y\in\mathcal{Y}} P_{\tilde{Y}}(y)^\alpha\right)^{\frac{1}{\alpha}} \lambda\right], \tag{2.76}$$

where

$$\lambda = \frac{\sum_{y\in\mathcal{Y}} \frac{P_Y(y)}{P_{\tilde{Y}}(y)} P_{\tilde{Y}}(y)^\alpha}{\sum_{y\in\mathcal{Y}} P_{\tilde{Y}}(y)^\alpha}. \tag{2.77}$$

Quadratic Loss

The quadratic loss function is $L_2(y,\hat{y}) = (y-\hat{y})^2$. The L-entropy function associated with the quadratic loss is the variance of Y, given by

$$H_2(Y) = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2. \tag{2.78}$$

The corresponding L-cross entropy is

$$H_2(Y;\tilde{Y}) = \mathbb{E}[Y^2] - 2\mathbb{E}[\tilde{Y}]\mathbb{E}[Y] + \mathbb{E}[\tilde{Y}]^2. \tag{2.79}$$

2.C Relationship among L-divergence, Bregman divergence, and f-divergence

We explain the relationship among the L-divergence defined in (2.41), the Bregman divergence [65], and the f-divergence [52]. All these three classes of divergence have been widely used in the machine learning literature. Their differences are explained below.

Let $\mathcal{P}^{\mathcal{Y}}$ denote the set of all probability distributions on the discrete set $\mathcal{Y}$. Define $\mathcal{Z} \subset \mathbb{R}^{|\mathcal{Y}|}$ as the set of all probability vectors $\mathbf{z} = (z_1, \ldots, z_{|\mathcal{Y}|})^T$ that satisfy $\sum_{i=1}^{|\mathcal{Y}|} z_i = 1$ and $z_i \geq 0$ for all $i = 1, 2, \ldots, |\mathcal{Y}|$. Any distribution $P_Y \in \mathcal{P}^{\mathcal{Y}}$ can be represented by a probability vector $\mathbf{p}_Y = (P_Y(y_1), \ldots, P_Y(y_{|\mathcal{Y}|}))^T \in \mathcal{Z}$.

Definition 2.3 *[65] Let $F : \mathcal{Z} \mapsto \mathbb{R}$ be a continuously differentiable and strictly convex function defined on the convex set $\mathcal{Z}$. The Bregman divergence associated with F between two distributions $P_Y, Q_Y \in \mathcal{P}^{\mathcal{Y}}$ is defined as*

$$B_F(P_Y||Q_Y) = F(\mathbf{p}_Y) - F(\mathbf{q}_Y) - \nabla F(\mathbf{q}_Y)^T(\mathbf{p}_Y - \mathbf{q}_Y), \tag{2.80}$$

where $\mathbf{p}_Y \in \mathcal{Z}$ and $\mathbf{q}_Y \in \mathcal{Z}$ are two probability vectors associated with the distributions P_Y and Q_Y, respectively, and $\nabla F = \left(\frac{\partial F}{\partial z_1}, \ldots, \frac{\partial F}{\partial z_{|\mathcal{Y}|}}\right)^T$.

We establish the following lemma:

Lemma 2.5 *For any continuously differentiable and strictly convex function $F : \mathcal{Z} \mapsto \mathbb{R}$, the Bregman divergence $B_F(P_Y||Q_Y)$ associated with F is an L_F-divergence $D_{L_F}(P_Y||Q_Y)$ associated with the loss function*

$$L_F(y, Q_Y) = -F(\mathbf{q}_Y) - \frac{\partial F(\mathbf{q}_Y)}{\partial z_y} + \nabla F(\mathbf{q}_Y)^T \mathbf{q}_Y. \tag{2.81}$$

Proof 2.11 *According to* (2.5)*, the L_F-entropy associated with the loss function $L_F(y, Q_Y)$ in* (2.81) *is defined as*

$$H_{L_F}(Y) = \min_{Q_Y \in \mathcal{P}^{\mathcal{Y}}} E_{Y \sim P_Y} \left[L_F \left(Y, Q_Y \right) \right], \tag{2.82}$$

where P_Y is the distribution of Y. Using (2.81)*, we can get*

$$\begin{aligned} E_{Y \sim P_Y} \left[L_F \left(Y, Q_Y \right) \right] &= -\sum_{y \in \mathcal{Y}} P_Y(y) F(\mathbf{q}_Y) - \sum_{y \in \mathcal{Y}} P_Y(y) \frac{\partial F(\mathbf{q}_Y)}{\partial z_y} \\ &\quad + \sum_{y \in \mathcal{Y}} P_Y(y) \nabla F(\mathbf{q}_Y)^T \mathbf{q}_Y \\ &= -F(\mathbf{q}_Y) - \nabla F(\mathbf{q}_Y)^T (\mathbf{p}_Y - \mathbf{q}_Y), \end{aligned} \tag{2.83}$$

where the last equality holds because $F(\mathbf{q}_Y)$ and $\nabla F(\mathbf{q}_Y)^T \mathbf{q}_Y$ are constants that remain unchanged regardless of the variable y. Because the function F is continuously differentiable and strictly convex on the convex set $\mathcal{Z}$, we have for all $\mathbf{p}_Y, \mathbf{q}_Y \in \mathcal{Z}$

$$-F(\mathbf{q}_Y) - \nabla F(\mathbf{q}_Y)^T (\mathbf{p}_Y - \mathbf{q}_Y) \geq -F(\mathbf{p}_Y). \tag{2.84}$$

Equality holds in (2.84) *if and only if $\mathbf{q}_Y = \mathbf{p}_Y$. From* (2.82)*,* (2.83)*, and* (2.84)*, we obtain*

$$\begin{aligned} H_{L_F}(Y) &= \min_{\mathbf{q}_Y \in \mathcal{Z}} -F(\mathbf{q}_Y) - \nabla F(\mathbf{q}_Y)^T (\mathbf{p}_Y - \mathbf{q}_Y) \\ &= -F(\mathbf{p}_Y). \end{aligned} \tag{2.85}$$

Substituting (2.83) *and* (2.85) *into* (2.80)*, yields*

$$\begin{aligned} B_F(P_Y || Q_Y) &= F(\mathbf{p}_Y) - F(\mathbf{q}_Y) - \nabla F(\mathbf{q}_Y)^T (\mathbf{p}_Y - \mathbf{q}_Y) \\ &= E_{Y \sim P_Y} \left[L_F \left(Y, Q_Y \right) \right] - H_{L_F}(Y) \\ &= D_{L_F}(P_Y || Q_Y). \end{aligned} \tag{2.86}$$

This completes the proof.

Let us rewrite the L-entropy $H_L(Y)$ as $H_L(\mathbf{p}_Y)$ to emphasize that it is a function of the probability vector $\mathbf{p}_Y$. If $H_L(\mathbf{p}_Y)$ is continuously differentiable and strictly concave in $\mathbf{p}_Y$, then the L-divergence $D_L(P_Y||Q_Y)$ can be expressed as [47, Section 3.5.4]

$$\begin{aligned} D_L(P_Y||Q_Y) &= H_L(\mathbf{q}_Y) + \nabla H_L(\mathbf{q}_Y)^T(\mathbf{p}_Y - \mathbf{q}_Y) - H_L(\mathbf{p}_Y) \\ &= B_{-H_L}(P_Y||Q_Y), \end{aligned} \tag{2.87}$$

which is the Bregman divergence $B_{-H_L}(P_Y||Q_Y)$ associated with the continuously differentiable and strictly convex $-H_L$. However, if $H_L(\mathbf{p}_Y)$ is not continuously differentiable or not strictly concave in $\mathbf{p}_Y$, $D_L(P_Y||Q_Y)$ is not necessarily a Bregman divergence.

Definition 2.4 *[52] Let $f : (0, \infty) \mapsto \mathbb{R}$ be a convex function with $f(1) = 0$. The f-divergence between two probability distributions $P_Y, Q_Y \in \mathcal{P}^{\mathcal{Y}}$ is defined as*

$$D_f(P_Y||Q_Y) = \sum_{y \in \mathcal{Y}} Q_Y(y) f\left(\frac{P_Y(y)}{Q_Y(y)}\right). \tag{2.88}$$

An f-divergence may not be L-divergence, and vice versa. In fact, the KL divergence $D_{\log}(P_Y||Q_Y)$ defined in (2.19) and its dual $D_{\log}(Q_Y||P_Y)$ are the unique divergences belonging to both the classes of f-divergence and Bregman divergence [66]. Because KL divergence is also an L-divergence, $D_{\log}(P_Y||Q_Y)$ and $D_{\log}(Q_Y||P_Y)$ are the only divergences belonging to all the three classes of divergences.

The f-mutual information can be expressed using the f-divergence as

$$I_f(Y; X) = \mathbb{E}_{X \sim P_X}[D_f(P_{Y|X}||P_Y)]. \tag{2.89}$$

The f-mutual information is symmetric, i.e., $I_f(Y;X) = I_f(X;Y)$, which can be shown as follows:

$$\begin{aligned}
I_f(Y;X) =& \sum_{x\in\mathcal{X}} P_X(x) \sum_{y\in\mathcal{Y}} P_Y(y) f\left(\frac{P_{Y|X}(y|x)}{P_Y(y)}\right) \\
=& \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} P_X(x)P_Y(y) f\left(\frac{P_{Y|X}(y|x)P_X(x)}{P_Y(y)P_X(x)}\right) \\
=& \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} P_X(x)P_Y(y) f\left(\frac{P_{X|Y}(x|y)P_Y(y)}{P_Y(y)P_X(x)}\right) \\
=& \sum_{y\in\mathcal{Y}} P_Y(y) \sum_{x\in\mathcal{X}} P_X(x) f\left(\frac{P_{X|Y}(x|y)}{P_X(x)}\right) \\
=& I_f(X;Y). \qquad (2.90)
\end{aligned}$$

On the other hand, the L-mutual information is generally non-symmetric, i.e., $I_L(Y;X) \neq I_L(X;Y)$, except for some special cases. For example, Shannon's mutual information is defined by

$$I_{\log}(Y;X) = \mathbb{E}_{X\sim P_X}[D_{\log}(P_{Y|X}||P_Y)], \qquad (2.91)$$

which is both an L-mutual information and a f-mutual information. It is well-known that $I_{\log}(Y;X) = I_{\log}(X;Y)$.

2.D Proof of Equation (2.3)

Because we assume that Y_t and $X_{t-\delta}$ are independent of $\Delta(t)$, for all $y \in \mathcal{Y}$, $x \in \mathcal{X}$, and $\delta \in \mathbb{Z}^+$, we have

$$P_{Y_t,X_{t-\delta}|\Delta(t)=\delta}(y,x) = P_{Y_t,X_{t-\delta}}(y,x). \qquad (2.92)$$

By using ϕ^* on the inference dataset, the inference error given $\Delta(t) = \delta$ is determined by

$$p(\delta) = \mathbb{E}_{Y,X\sim P_{Y_t,X_{t-\Delta(t)}|\Delta(t)=\delta}}\left[L\left(Y,\phi^*(X,\delta)\right)\right], \qquad (2.93)$$

where $P_{Y_t,X_{t-\Delta(t)}|\Delta(t)=\delta}$ is the distribution of target Y_t and feature $X_{t-\Delta(t)}$ given $\Delta(t)=\delta$.

By substituting (2.92) into (2.93), we obtain

$$\begin{aligned} p(\delta) &= \mathbb{E}_{Y,X\sim P_{Y_t,X_{t-\Delta(t)}|\Delta(t)=\delta}}\left[L\left(Y,\phi^*(X,\delta)\right)\right] \\ &= \mathbb{E}_{Y,X\sim P_{Y_t,X_{t-\delta}|\Delta(t)=\delta}}\left[L\left(Y,\phi^*(X,\delta)\right)\right] \\ &= \mathbb{E}_{Y,X\sim P_{Y_t,X_{t-\delta}}}\left[L\left(Y,\phi^*(X,\delta)\right)\right] \\ &= \mathbb{E}_{Y,X\sim P_{Y_\delta,X_0}}\left[L\left(Y,\phi^*(X,\delta)\right)\right]. \end{aligned} \tag{2.94}$$

The last equality holds due to the stationarity of $\{(Y_t,X_t), t=0,1,2,\ldots\}$. This completes the proof.

2.E Proof of Equation (2.10)

Let $\mathcal{D}=\{\delta : P_\Theta(\delta)>0\}$ be support set of P_Θ. From (2.9), we have

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta},\Theta) &= \sum_{x\in\mathcal{X},\delta\in\mathcal{D}} P_{\tilde{X}_{-\Theta},\Theta}(x,\delta) H_L(\tilde{Y}_0|\tilde{X}_{-\Theta}=x,\Theta=\delta) \\ &= \sum_{\delta\in\mathcal{D}} P_\Theta(\delta) \sum_{x\in\mathcal{X}} P_{\tilde{X}_{-\Theta}|\Theta=\delta}(x) H_L(\tilde{Y}_0|\tilde{X}_{-\Theta}=x,\Theta=\delta) \\ &= \sum_{\delta\in\mathcal{D}} P_\Theta(\delta) \sum_{x\in\mathcal{X}} P_{\tilde{X}_{-\delta}|\Theta=\delta}(x) H_L(\tilde{Y}_0|\tilde{X}_{-\Theta}=x,\Theta=\delta). \end{aligned} \tag{2.95}$$

Next, from (2.6), we obtain that for all $x\in\mathcal{X}$ and $\delta\in\mathcal{D}$,

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta}=x,\Theta=\delta) &= \min_{a\in\mathcal{A}} \mathbb{E}_{Y\sim P_{\tilde{Y}_0|\tilde{X}_{-\Theta}=x,\Theta=\delta}}[L(Y,a)] \\ &= \min_{a\in\mathcal{A}} \mathbb{E}_{Y\sim P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x,\Theta=\delta}}[L(Y,a)]. \end{aligned} \tag{2.96}$$

Because we assume that $\tilde{Y}_0$ and $\tilde{X}_{-k}$ are independent of Θ for every $k\geq 0$, for all $x\in\mathcal{X}, y\in\mathcal{Y}$ and $\delta\in\mathcal{D}$

$$P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x,\Theta=\delta}(y) = P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}(y), \tag{2.97}$$

$$P_{\tilde{X}_{-\delta}|\Theta=\delta}(x) = P_{\tilde{X}_{-\delta}}(x). \tag{2.98}$$

Substituting (2.97) into (2.96), we get

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta} = x, \Theta = \delta) &= \min_{a\in\mathcal{A}} \mathbb{E}_{Y\sim P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}}[L(Y,a)] \\ &= H_L(\tilde{Y}_0|\tilde{X}_{-\delta} = x). \end{aligned} \tag{2.99}$$

Substituting (2.99) and (2.98) into (2.95), we observe that

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta}, \Theta) &= \sum_{\delta\in\mathcal{D}} P_\Theta(\delta) \sum_{x\in\mathcal{X}} P_{\tilde{X}_{-\delta}}(x) H_L(\tilde{Y}_0|\tilde{X}_{-\delta} = x) \\ &= \sum_{\delta\in\mathbb{Z}^+} P_\Theta(\delta)\ H_L(\tilde{Y}_0|\tilde{X}_{-\delta}). \end{aligned} \tag{2.100}$$

This completes the proof.

2.F Proof of Lemma 2.1

This is due to the following symmetry property:

$$I_{\log}(Y; Z|X) = I_{\log}(Z; Y|X). \tag{2.101}$$

2.G Proof of Lemma 2.2

By using the definition of L-conditional mutual information in (2.44), we obtain

$$\begin{aligned} H_L(Y|X,Z) =& H_L(Y|X) - I_L(Y; Z|X) \\ =& H_L(Y|Z) - I_L(Y; X|Z). \end{aligned} \tag{2.102}$$

From (2.44) and (2.102), we get

$$\begin{aligned} H_L(Y|X) =& H_L(Y|Z) + I_L(Y; Z|X) - I_L(Y; X|Z) \\ \leq& H_L(Y|Z) + I_L(Y; Z|X), \end{aligned} \tag{2.103}$$

where the last inequality is due to $I_L(Y;X|Z) \geq 0$. Now, we need to show that if $Y \overset{\epsilon}{\leftrightarrow} X \overset{\epsilon}{\leftrightarrow} Z$, then

$$I_L(Y;Z|X) = O(\epsilon). \tag{2.104}$$

and in addition, if $H_L(Y)$ is twice differentiable, then

$$I_L(Y;Z|X) = O(\epsilon^2). \tag{2.105}$$

From (2.44), we see that

$$I_L(Y;Z|X) = \mathbb{E}_{X,Z}[D_L(P_{Y|X,Z}||P_{Y|X})]. \tag{2.106}$$

We know from Pinsker's inequality [55, Lemma 11.6.1] that

$$\sum_{y\in\mathcal{Y}}(P_Y(y) - Q_Y(y))^2 \leq 2\ln 2 \; D_{\log}(P_Y||Q_Y). \tag{2.107}$$

If $Y \overset{\epsilon}{\leftrightarrow} X \overset{\epsilon}{\leftrightarrow} Z$ is an ϵ-Markov chain, then Definition 2.1 yields:

$$\sum_{(x,z)\in\mathcal{X}\times\mathcal{Z}} P_{X,Z}(x,z) D_{\log}(P_{Y|X=x,Z=z}||P_{Y|X=x}) \leq \epsilon^2. \tag{2.108}$$

Let $\mathcal{X}' \times \mathcal{Z}' = \{(x,z) : P_{X,Z}(x,z) > 0\}$ be the support set of $P_{X,Z}$. Then, (2.109) reduces to

$$\sum_{(x,z)\in\mathcal{X}'\times\mathcal{Z}'} P_{X,Z}(x,z) D_{\log}(P_{Y|X=x,Z=z}||P_{Y|X=x}) \leq \epsilon^2. \tag{2.109}$$

Because the left side of the above inequality is the summation of non-negative terms, the following holds:

$$P_{X,Z}(x,z) D_{\log}(P_{Y|X=x,Z=z}||P_{Y|X=x}) \leq \epsilon^2, \tag{2.110}$$

for all $(x,z) \in \mathcal{X}' \times \mathcal{Z}'$. Because $P_{X,Z}(x,z) > 0$ for all $(x,z) \in \mathcal{X}' \times \mathcal{Z}'$, from (2.107) and (2.110), we can write

$$\sum_{y \in \mathcal{Y}} (P_{Y|X=x,Z=z}(y) - P_{Y|X=x}(y))^2 \leq \frac{2\ln 2 \epsilon^2}{P_{X,Z}(x,z)}, \tag{2.111}$$

for all $(x,z) \in \mathcal{X}' \times \mathcal{Z}'$. Next, we need the following lemma.

Lemma 2.6 *The following assertions are true:*

(a) If two distributions $Q_Y \in \mathcal{P}^{\mathcal{Y}}$ and $P_Y \in \mathcal{P}^{\mathcal{Y}}$ satisfy

$$\sum_{y \in \mathcal{Y}} (P_Y(y) - Q_Y(y))^2 \leq \beta^2, \tag{2.112}$$

then

$$D_L(P_Y||Q_Y) = O(\beta). \tag{2.113}$$

(b) If, in addition, $H_L(Y)$ is twice differentiable in P_Y, then

$$D_L(P_Y||Q_Y) = O(\beta^2). \tag{2.114}$$

Proof 2.12 *See in Appendix 2.L.*

Using (2.111) and Lemma 2.6(a) in (2.106), we obtain

$$\begin{aligned} I_L(Y;Z|X) &= \sum_{(x,z) \in \mathcal{X} \times \mathcal{Z}} P_{X,Z}(x,z)\ D_L(P_{Y|X=x,Z=z}||P_{Y|X=x}) \\ &= \sum_{(x,z) \in \mathcal{X}' \times \mathcal{Z}'} P_{X,Z}(x,z)\ D_L(P_{Y|X=x,Z=z}||P_{Y|X=x}) \\ &= \sum_{(x,z) \in \mathcal{X}' \times \mathcal{Z}'} P_{X,Z}(x,z)\ O\left(\frac{\sqrt{2\ln 2}\epsilon}{\sqrt{P_{X,Z}(x,z)}}\right) \\ &= O(\epsilon). \end{aligned} \tag{2.115}$$

Similarly, when $H_L(Y)$ is differentiable in P_Y, by using Lemma 2.6(b) we obtain

$$I_L(Y;Z|X) = O(\epsilon^2). \tag{2.116}$$

This completes the proof of Lemma 2.2.

2.H Proof of Theorem 2.1

By using the definition of L-conditional mutual information in (2.44), we can show that

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-k},\tilde{X}_{-k-1}) &= H_L(\tilde{Y}_0|\tilde{X}_{-k-1}) - I_L(\tilde{Y}_0;\tilde{X}_{-k}|\tilde{X}_{-k-1}) \\ &= H_L(\tilde{Y}_0|\tilde{X}_{-k}) - I_L(\tilde{Y}_0;\tilde{X}_{-k-1}|\tilde{X}_{-k}). \end{aligned} \tag{2.117}$$

Expanding $H_L(\tilde{Y}_0|\tilde{X}_{-k})$, we have

$$H_L(\tilde{Y}_0|\tilde{X}_{-k}) = H_L(\tilde{Y}_0|\tilde{X}_{-k-1}) + I_L(\tilde{Y}_0;\tilde{X}_{-k-1}|\tilde{X}_{-k}) - I_L(\tilde{Y}_0;\tilde{X}_{-k}|\tilde{X}_{-k-1}). \tag{2.118}$$

Since the above equation is valid for all values of $k \geq 0$, taking the summation of $H_L(\tilde{Y}_0|\tilde{X}_{-k})$ from $k = 0$ to $\delta - 1$ yields:

$$H_L(\tilde{Y}_0|\tilde{X}_{-\delta}) = H_L(\tilde{Y}_0|\tilde{X}_0) + \sum_{k=0}^{\delta-1} I_L(\tilde{Y}_0;\tilde{X}_{-k}|\tilde{X}_{-k-1}) - \sum_{k=0}^{\delta-1} I_L(\tilde{Y}_0;\tilde{X}_{-k-1}|\tilde{X}_{-k}). \tag{2.119}$$

Thus, we can express $H_L(\tilde{Y}_0|\tilde{X}_{-\delta})$ as a function of δ as in (2.23) and (2.24). Moreover, the functions $g_1(\delta)$ and $g_2(\delta)$ defined in (2.24) are non-decreasing in δ as $I_L(\tilde{Y}_0;\tilde{X}_{-k}|\tilde{X}_{-k-1}) \geq 0$ and $I_L(\tilde{Y}_0;\tilde{X}_{-k-1}|\tilde{X}_{-k}) \geq 0$ for all values of k.

To prove the next part, we use Lemma 2.2. Because for every $\mu, \nu \geq 0$, $\tilde{Y}_0 \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu} \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu-\nu}$ is an ϵ-Markov chain, we can write

$$I_L(\tilde{Y}_0;\tilde{X}_{-k-1}|\tilde{X}_{-k}) = O(\epsilon). \tag{2.120}$$

This implies

$$g_2(\delta) = \sum_{k=0}^{\delta-1} O(\epsilon) = O(\epsilon). \tag{2.121}$$

The last equality follows from the summation property of big-O-notation. This completes the proof.

2.I Proof of Theorem 2.2

Using (2.10) and Theorem 2.1, we obtain

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta}, \Theta) &= \sum_{\delta \in \mathbb{Z}^+} P_\Theta(\delta) \big(H_L(\tilde{Y}_0|\tilde{X}_0) + \hat{g}_1(\delta) - g_2(\delta) \big) \\ &= H_L(\tilde{Y}_0|\tilde{X}_0) + \mathbb{E}_{\Theta \sim P_\Theta}[\hat{g}_1(\Theta)] - \mathbb{E}_{\Theta \sim P_\Theta}[g_2(\Theta)], \end{aligned} \tag{2.122}$$

where

$$\begin{aligned} \hat{g}_1(\delta) &= g_1(\delta) - H_L(\tilde{Y}_0|\tilde{X}_0) \\ &= \sum_{k=0}^{\delta-1} I_L(\tilde{Y}_0; \tilde{X}_{-k}|\tilde{X}_{-k-1}). \end{aligned} \tag{2.123}$$

Because mutual information $I_L(\tilde{Y}_0; \tilde{X}_{-k}|\tilde{X}_{-k-1})$ is non-negative, we have

$$\hat{g}_1(\delta) = \sum_{k=0}^{\delta-1} I_L(\tilde{Y}_0; \tilde{X}_{-k}|\tilde{X}_{-k-1}) \geq 0. \tag{2.124}$$

Because $\hat{g}_1(\delta)$ is non-negative for all δ, the function $\hat{g}_1(\cdot)$ is Lebesgue integrable with respect to all probability measure P_Θ [67]. Hence, the expectation $\mathbb{E}_{\Theta \sim P_\Theta}[\hat{g}_1(\Theta)]$ exists. Note that $E_{\Theta \sim P_\Theta}[\hat{g}_1(\Theta)]$ can be infinite $(+\infty)$. By using the same argument, we obtain that $E_{\Theta \sim P_\Theta}[g_2(\Theta)]$ exists but can also be infinite. Moreover, the functions $\hat{g}_1(\delta)$ and $g_2(\delta)$ are non-decreasing in δ.

Because (i) the function $\hat{g}_1(\delta)$ is non-decreasing in δ, (ii) the expectation $\mathbb{E}_{\Theta\sim P_\Theta}[\hat{g}_1(\Theta)]$ exists, and (iii) $\Theta_1 \leq_{st} \Theta_2$, we have [56]

$$\mathbb{E}_{\Theta\sim P_{\Theta_1}}[\hat{g}_1(\Theta)] \leq \mathbb{E}_{\Theta\sim P_{\Theta_2}}[\hat{g}_1(\Theta)]. \tag{2.125}$$

Next, we obtain:

$$\begin{aligned} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta_1},\Theta_1) &\overset{(a)}{=} H_L(\tilde{Y}_0|\tilde{X}_0) + \mathbb{E}_{\Theta\sim P_{\Theta_1}}[\hat{g}_1(\Theta)] - \mathbb{E}_{\Theta\sim P_{\Theta_1}}[g_2(\Theta)] \\ &\overset{(b)}{\leq} H_L(\tilde{Y}_0|\tilde{X}_0) + \mathbb{E}_{\Theta\sim P_{\Theta_2}}[\hat{g}_1(\Theta)] - \mathbb{E}_{\Theta\sim P_{\Theta_1}}[g_2(\Theta)] \\ &\overset{(c)}{=} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta_2},\Theta_2) \\ &\quad + \mathbb{E}_{\Theta\sim P_{\Theta_2}}[g_2(\Theta)] - \mathbb{E}_{\Theta\sim P_{\Theta_1}}[g_2(\Theta)] \\ &\overset{(d)}{=} H_L(\tilde{Y}_0|\tilde{X}_{-\Theta_2},\Theta_2) + O(\epsilon), \end{aligned} \tag{2.126}$$

where (a) and (c) hold due to (2.122), (b) is obtained using (2.125), and (d) follows from the fact that $\tilde{Y}_0 \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu} \overset{\epsilon}{\leftrightarrow} \tilde{X}_{-\mu-\nu}$ is an ϵ-Markov chain for all $\mu,\nu \geq 0$ (see Theorem 2.1(b)). This completes the proof.

2.J Proof of Lemma 2.3

By using condition (2.28) and Lemma 2.6(a), we obtain for all $x \in \mathcal{X}$:

$$D_L\left(P_{Y_t|X_{t-\delta}=x}||P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}\right) = O(\beta). \tag{2.127}$$

Next, by using (2.11) and (2.127), we have

$$\begin{aligned} H_L(P_{Y_t|X_{t-\delta}};P_{\tilde{Y}_0|X_{-\delta}}|P_{X_{t-\delta}}) &= H_L(Y_t|X_{t-\delta}) \\ &\quad + \sum_{x\in\mathcal{X}} P_{X_{t-\delta}}(x)\ D_L\left(P_{Y_t|X_{t-\delta}=x}||P_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}\right) \\ &= H_L(Y_t|X_{t-\delta}) + O(\beta). \end{aligned} \tag{2.128}$$

This completes the proof.

2.K Proof of Theorem 2.3

Part (a): By the definition of L-conditional cross entropy (2.11), we get

$$H_L(P_{Y_t|X_{t-\delta}}; P_{\tilde{Y}_0|X_{-\delta}}|P_{X_{t-\delta}}) = \sum_{x\in\mathcal{X}} P_{X_{t-\delta}}(x)\mathbb{E}_{Y\sim P_{Y_t|X_{t-\delta}=x}}\left[L\left(Y, a_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}\right)\right], \tag{2.129}$$

where the Bayes predictor $a_{\tilde{Y}_0|\tilde{X}_{-\delta}=x}$ is fixed in the inference phase for every time slot t. Because $\{(Y_t, X_t), t = 0, 1, 2, \ldots\}$ is a stationary process, (2.129) is a function of the AoI δ.

Part (b): We can apply Lemma 2.3 since (2.28) holds for all $x \in \mathcal{X}$ and $\delta \in \mathcal{Z}^+$. This gives us:

$$\begin{aligned}
&H_L(P_{Y_t|X_{t-\delta_1}}; P_{\tilde{Y}_0|X_{-\delta_1}}|P_{X_{t-\delta_1}}) \\
=&H_L(Y_t|X_{t-\delta_1}) + O(\beta) \\
\leq&H_L(Y_t|X_{t-\delta_2}) + O(\epsilon) + O(\beta) \\
=&H_L(P_{Y_t|X_{t-\delta_2}}; P_{\tilde{Y}_0|X_{-\delta_2}}|P_{X_{t-\delta_2}}) + O(\beta) + O(\epsilon) + O(\beta) \\
=&H_L(P_{Y_t|X_{t-\delta_2}}; P_{\tilde{Y}_0|X_{-\delta_2}}|P_{X_{t-\delta_2}}) + O(\max\{\epsilon, \beta\}),
\end{aligned} \tag{2.130}$$

where we use Lemma 2.3 to obtain the first and the third equality, and the second inequality holds due to the assumption that $Y_t \overset{\epsilon}{\leftrightarrow} X_{t-\mu} \overset{\epsilon}{\leftrightarrow} X_{t-\mu-\nu}$ is an ϵ-Markov chain for all $\mu, \nu \geq 0$ (see Theorem 2.1). This completes the proof.

2.L Proof of Lemma 2.6

To prove Lemma 2.6, we will use the sub-gradient mean value theorem [68]. When $H_L(Y)$ is twice differentiable in P_Y, we can use a second order Taylor series expansion.

If (2.112) holds, we can obtain

$$\sum_{y\in\mathcal{Y}} (P_Y(y) - Q_Y(y))^2 \leq \beta^2, \tag{2.131}$$

$$\sum_{y\in\mathcal{Y}} |P_Y(y) - Q_Y(y)| \leq \beta, \tag{2.132}$$

and

$$\max_{y\in\mathcal{Y}} |P_Y(y) - Q_Y(y)| \leq \beta. \tag{2.133}$$

Let us define a convex function $g : \mathbb{R}^{|\mathcal{Y}|} \mapsto \mathbb{R}$ as

$$g(\mathbf{z}) = \sum_{i=1}^{|\mathcal{Y}|} z_i \; L(y_i, a_{Q_Y}) - \min_{a\in\mathcal{A}} \sum_{i=1}^{|\mathcal{Y}|} z_i \; L(y_i, a), \tag{2.134}$$

where a_{Q_Y} is a Bayes action associated with the distribution Q_Y, i.e., a_{Q_Y} is the minimizer of

$$a_{Q_Y} = \operatorname*{argmin}_{a\in\mathcal{A}} \mathbb{E}_{Y\sim Q_Y}[L(Y, a]]. \tag{2.135}$$

Because $g(\mathbf{z})$ is a convex function and the set of sub-gradients of $g(\mathbf{z})$ is bounded [68, Proposition 4.2.3], we can apply the sub-gradient mean value theorem [68] along with (2.41), and (2.132) to obatin

$$\begin{aligned} g(\mathbf{p}_Y) =& D_L(P_Y||Q_Y) \\ =& g(\mathbf{q}_Y) + O\left(\sum_{y\in\mathcal{Y}} |P_Y(y) - Q_Y(y)|\right) \\ =& D_L(Q_Y||Q_Y) + O\left(\sum_{y\in\mathcal{Y}} |P_Y(y) - Q_Y(y)|\right) \\ =& O(\beta). \end{aligned} \tag{2.136}$$

Now, let us consider the case where $H_L(Y)$ is assumed to be twice differentiable in P_Y. The function $g(\mathbf{p}_Y)$ can also be expressed in terms of $H_L(Y)$ as:

$$\begin{aligned} g(\mathbf{p}_Y) &= \sum_{i=1}^{|\mathcal{Y}|} P_Y(y_i)\ L(y_i, a_{Q_Y}) - \min_{a \in \mathcal{A}} \sum_{i=1}^{|\mathcal{Y}|} P_Y(y_i)\ L(y_i, a) \\ &= \sum_{i=1}^{|\mathcal{Y}|} P_Y(y_i)\ L(y_i, a_{Q_Y}) - H_L(Y). \end{aligned} \tag{2.137}$$

Because $H_L(Y)$ is assumed to be twice differentiable in P_Y, from (2.137), we can conclude that $g(\mathbf{p}_Y)$ is twice differentiable with respect to $\mathbf{p}_Y$. Furthermore, we have

$$g(\mathbf{p}_Y) \geq 0, \forall \mathbf{p}_Y \in \mathbb{R}^{|\mathcal{Y}|} \tag{2.138}$$

and

$$g(\mathbf{q}_Y) = D_L(Q_Y||Q_Y) = 0. \tag{2.139}$$

Using the first-order necessary condition for optimality, we find that the gradient of $g(\mathbf{p}_Y)$ at point $\mathbf{p}_Y = \mathbf{q}_Y$ is zero, i.e.,

$$\nabla g(\mathbf{q}_Y) = 0. \tag{2.140}$$

Next, based on (2.139) and (2.140), we can perform a second-order Taylor series expansion of $g(\mathbf{p}_Y)$ at $\mathbf{p}_Y = \mathbf{q}_Y$:

$$\begin{aligned} g(\mathbf{p}_Y) =& g(\mathbf{q}_Y) + (\mathbf{p}_Y - \mathbf{q}_Y)^T \nabla g(\mathbf{q}_Y) \\ &+ \frac{1}{2}(\mathbf{p}_Y - \mathbf{q}_Y)^T \mathcal{H}(\mathbf{q}_Y)(\mathbf{p}_Y - \mathbf{q}_Y) \\ &+ o\left(\sum_{y\in\mathcal{Y}} (P_Y(y) - Q_Y(y))^2\right) \\ =& \frac{1}{2}(\mathbf{p}_Y - \mathbf{q}_Y)^T \mathcal{H}(\mathbf{q}_Y)(\mathbf{p}_Y - \mathbf{q}_Y) \\ &+ o\left(\sum_{y\in\mathcal{Y}} (P_Y(y) - Q_Y(y))^2\right), \end{aligned} \tag{2.141}$$

where $\mathcal{H}(\mathbf{q}_Y)$ is the Hessian matrix of $g(\mathbf{p}_Y)$ at point $\mathbf{p}_Y = \mathbf{q}_Y$.

Because $g(\mathbf{p}_Y)$ is a convex function, we have

$$(\mathbf{p}_Y - \mathbf{q}_Y)^T \mathcal{H}(\mathbf{q}_Y)(\mathbf{p}_Y - \mathbf{q}_Y) \geq 0.$$

Moreover, we can express

$$\begin{aligned} &\frac{1}{2}(\mathbf{p}_Y - \mathbf{q}_Y)^T \mathcal{H}(\mathbf{q}_Y)(\mathbf{p}_Y - \mathbf{q}_Y) \\ &= \frac{1}{2}\sum_{y,y'} (P_Y(y) - Q_Y(y))\mathcal{H}(\mathbf{q}_Y)_{y,y'}(P_Y(y') - Q_Y(y')) \\ &= O\left(\sum_{y,y'} (P_Y(y) - Q_Y(y))(P_Y(y') - Q_Y(y'))\right). \end{aligned} \tag{2.142}$$

Now, by substituting (2.142) into (2.141), we obtain

$$\begin{aligned} g(\mathbf{p}_Y) =& O\left(\sum_{y,y'} (P_Y(y) - Q_Y(y))(P_Y(y') - Q_Y(y'))\right) \\ &+ o\left(\sum_{y\in\mathcal{Y}} (P_Y(y) - Q_Y(y))^2\right). \end{aligned} \tag{2.143}$$

Using (2.131) and (2.133), we obtain from (2.143) that

$$g(\mathbf{p}_Y) = D_L(P_Y||Q_Y) = O(\beta^2) + o(\beta^2) = O(\beta^2). \tag{2.144}$$

This completes the proof.

2.M Proof of Lemma 2.4

Because $Y_t = f(X_{t-d})$ and X_t is a Markov chain, $Y_t \leftrightarrow X_{t-\delta} \leftrightarrow X_{t-(\delta-1)}$ is a Markov chain for all $0 \leq \delta \leq d$. By the data processing inequality for L-conditional entropy [48, Lemma 12.1], one can show that for all $0 \leq \delta \leq d$,

$$H_L(Y_t|X_{t-\delta}) \leq H_L(Y_t|X_{t-(\delta-1)}). \tag{2.145}$$

Moreover, since $Y_t = f(X_{t-d})$ and X_t is a Markov chain, $Y_t \leftrightarrow X_{t-\delta} \leftrightarrow X_{t-(\delta+1)}$ is a Markov chain for all $\delta \geq d$. By the data processing inequality [48, Lemma 12.1], one can show that for all $\delta \geq d$,

$$H_L(Y_t|X_{t-\delta}) \leq H_L(Y_t|X_{t-(\delta+1)}). \tag{2.146}$$

Because $Y_t = f(X_{t-d})$ and $f(\cdot)$ is a function, we have $P_{Y_t|X_{t-d}} = P_{Y_t|f(X_{t-d})} = P_{Y_t|Y_t}$. Hence,

$$H_L(Y_t|X_{t-d}) = H_L(Y_t|Y_t) \leq H_L(Y_t|Z). \tag{2.147}$$

Chapter 3

Timely Inference in Single-Source and Single-Channel Networks

3.1 Introduction

In this chapter, we will introduce a new medium access model and develop a novel transmission scheduling policy for single-source remote inference systems. This scheduling policy can effectively minimize general functions of the AoI, regardless of whether the function is monotonic or not. The contributions of this paper are summarized as follows:

- We design transmission scheduling policies for minimizing the inference error. Because fresher data is not always better, we propose a new medium access model called the "selection-from-buffer" model, where B most recent features are stored in the source's buffer and the source can choose to send any of the B most recent features. This model is more general than the "generate-at-will" model used in earlier studies, e.g., [20–25, 28]. If the inference error is an non-decreasing function of the AoI, the "selection-from-buffer" model achieves the same performance as the "generate-at-will" model; if the AoI function is non-monotonic, the "selection-from-buffer" model can potentially achieve better performance.

- When there is a single source-predictor pair and a single channel, an optimal scheduling policy is devised to determine (i) when to submit features to the channel and (ii) which feature in the buffer to submit. This scheduling policy is capable of minimizing general functions of the AoI, regardless of whether the function is monotonic or not. By leveraging a new index function $\gamma(\Delta(t))$, the optimal scheduling policy can be expressed an *index-based threshold policy*, where a new packet is sent out whenever $\gamma(\Delta(t))$ exceeds a pre-determined threshold (Theorems 3.1-3.2). The threshold can be computed by using low complexity algorithms, e.g., bisection search. We note that the function $\gamma(\cdot)$ is *not necessarily monotonic* and hence its inverse function *may not exist*. Consequently, this index-based threshold policy cannot be equivalently expressed as an AoI-based threshold policy, i.e., a new packet is sent out whenever the AoI $\Delta(t)$

exceeds a pre-determined threshold. This is a key difference from prior studies on minimizing non-decreasing AoI functions [20–32], where the optimal scheduling policy is an AoI-based threshold policy.

- The above results hold (i) for minimizing general AoI functions (monotonic or non-monotonic) and (ii) for random delay channels. Data-driven evaluations show that the optimal scheduler achieves up to 3 times smaller inference error compared to "generate-at-will" with optimal scheduling strategy and 8 times smaller inference error compared to periodic feature updating (see Fig. 3.2).

- When the training and inference data have the same probabilistic distribution, remote inference reduces to signal-agnostic remote estimation. Hence, the results of the present work above also apply to signal-agnostic remote estimation.

3.1.1 Related Works

In the earlier AoI studies [20–32], it was usually assumed that the observed data sequence is Markovian and the performance degradation caused by information aging was modeled as a monotonic AoI function. Hence, the earlier studies [20–25, 28] adopted "generate-at-will" status updating model, where the transmitter can only select the most recently generated signal. However, practical data sequence may not be Markovian [23]. In the present paper, we propose a new local geometric approach to analyze both Markovian and non-Markovian time-series data. For non-Markovian time-series data, fresh data is not always better. To that end, we propose a new status updating model called the "selection-from-buffer" model, where the transmitter has the option to send any of the B most recent features stored in the source buffer.

3.2 Selection-from-Buffer: A New Status Updating Model

Consider the remote inference system depicted in Fig. 3.1, where a source progressively generates features and sends them through a channel to a receiver. The system operates in discrete time-slots. In each time slot t, a pre-trained neural network at the receiver employs the freshest received feature $X_{t-\Delta(t)}$ to infer the current label Y_t.

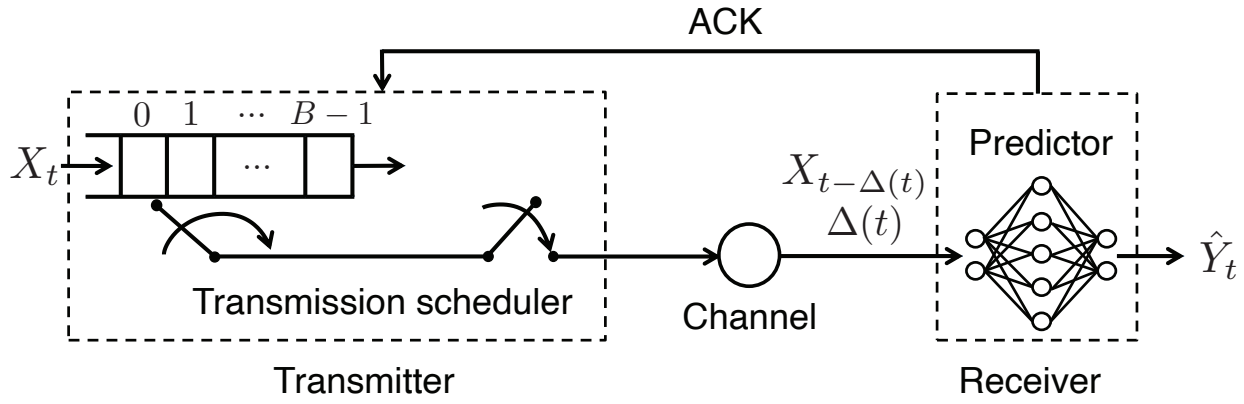


Figure 3.1: A remote inference system with "selection-from-buffer." At each time slot t, the transmitter generates a feature X_t and keeps it in a buffer that stores B most recent features $(X_t, X_{t-1}, \ldots, X_{t-B+1})$. The scheduler decides (i) when to submit features to the channel and (ii) which feature in the buffer to submit.

As discussed in Section 2.2-2.4, the inference error $\text{err}_{\text{inference}}(\Delta(t))$ is a function of the AoI $\Delta(t)$, whereas the function is not necessarily monotonically increasing. In certain scenarios, a stale feature with $\Delta(t) > 0$ can outperform a freshly generated feature with $\Delta(t) = 0$. Inspired by these observations, we introduce a novel medium access model for status updating, which is termed the "selection-from-buffer" model. In this model, the source maintains a buffer that stores the B most recent features $(X_t, X_{t-1}, \ldots, X_{t-B+1})$ in each time slot t. Specifically, at the beginning of time slot t, the source appends a newly generated feature X_t to the buffer, while concurrently evicting the oldest feature X_{t-B}. If the channel is available at time slot t, the transmitter can send one of the B most recent features or remain silent, where the transmission may last for one or multiple time-slots. Notably, the "selection-from-buffer" model generalizes the "generate-at-will" model [20, 22], with the latter is a special case of the former with $B = 1$.[1]

The system starts to operate at time slot $t = 0$. We assume that the buffer is initially populated with B features $(X_0, X_{-1}, \ldots, X_{-B+1})$ at time slot $t = 0$. By this, the buffer is kept full at all time slot $t \geq 0$. The channel is modeled as a non-preemptive server with feature transmission times $T_i \geq 1$, which can be random due to factors like time-varying

[1]In comparison to the "generate-at-will" model, our "selection-from-buffer" model presents a critical advantage: it enables the systematic investigation of *optimal feature design for remote inference.* As an evidence, our subsequent study [34] demonstrates that substantial performance improvements can be attained through the joint optimization of transmission scheduling and the feature sequence length u.

channel conditions, collisions, random packet sizes, etc. We assume that the T_i's are *i.i.d.* with $1 \leq \mathbb{E}[T_i] < \infty$. The i-th feature is generated in time slot G_i, submitted to the channel in time slot S_i, and delivered to the receiver in time slot $D_i = S_i + T_i$, where $G_i \leq S_i < D_i$, and $D_i \leq S_{i+1} < D_{i+1}$. Once a feature is delivered, an acknowledgment (ACK) is fed back to the transmitter in the same time slot. Thus, the idle/busy state of the channel is known at the transmitter.

Scheduling Policies and Problem Formulation

A transmission scheduler determines (i) when to submit features to the channel and (ii) which feature in the buffer to submit. In time slot S_i, let $X_{G_i} = X_{S_i - b_i}$ be the feature submitted to the channel, which is the $(b_i + 1)$-th freshest feature in the buffer, with $b_i \in \{0, 1, \ldots, B-1\}$. By this, $G_i = S_i - b_i$. A scheduling policy is denoted by a 2-tuple (f, g), where $g = (S_1, S_2, \ldots)$ determines when to submit the features and $f = (b_1, b_2, \ldots)$ specifies which feature in the buffer to submit.

Let $U(t) = \max_i\{G_i : D_i \leq t\}$ represent the generation time of the freshest feature delivered to the receiver up to time slot t. Because $G_i = S_i - b_i$, $U(t) = \max_i\{S_i - b_i : D_i \leq t\}$. The age of information (AoI) at time t is [10]

$$\Delta(t) = t - U(t) = t - \max_i\{S_i - b_i : D_i \leq t\}. \tag{3.1}$$

Because $D_i < D_{i+1}$, $\Delta(t)$ can be re-written as

$$\Delta(t) = t - S_i + b_i, \quad \text{if} \quad D_i \leq t < D_{i+1}. \tag{3.2}$$

The initial state of the system is assumed to be $S_0 = 0, D_0 = T_0$, and $\Delta(0)$ is a finite constant.

We focus on the class of *signal-agnostic* scheduling policies in which each decision is determined without using the knowledge of the signal value of the observed process. A scheduling policy (f, g) is said to be signal-agnostic, if the policy is independent of $\{(Y_t, X_t), t = 0, 1, 2, \ldots\}$. Let Π represent the set of all causal and signal-agnostic scheduling policies that satisfy three conditions: (i) the transmission time schedule S_i and the buffer

position b_i are determined based on the current and the historical information available at the scheduler; (ii) the scheduler does not have access to the realization of the process $\{(Y_t, X_t), t = 0, 1, 2, \ldots\}$; and (iii) the scheduler can access the inference error function $\text{err}_{\text{inference}}(\cdot)$ and the distribution of T_i.

Our goal is to find an optimal scheduling policy that minimizes the time-average expected inference error among all causal scheduling policies in Π:

$$\bar{p}_{opt} = \inf_{(f,g)\in\Pi} \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{(f,g)}\left[\sum_{t=0}^{T-1} p(\Delta(t))\right]. \tag{3.3}$$

where we use a simpler notation $p(\Delta(t)) = \text{err}_{\text{inference}}(\Delta(t))$ to represent the inference error in time-slot t, and $\bar{p}_{opt}$ is the optimum value of (3.3). Because $p(\cdot)$ is not necessarily monotonic and the scheduler needs to determine which feature in the buffer to send, (3.3) is more challenging than the scheduling problems for minimizing non-decreasing age functions in [20–32]. Note that $p(\Delta(t)) = \text{err}_{\text{inference}}(\Delta(t))$ is the inference error in time-slot t, instead of its information-theoretic approximations.

3.3 An Optimal Scheduling Solution

To elucidate the optimal solution to the scheduling problem (3.3), we first fix the buffer position at $b_i = b$ for each feature i submitted to the channel and focus on the optimization of the transmission time schedule $g = (S_1, S_2, \ldots)$. This simplified transmission scheduling problem is expressed as

$$\bar{p}_{b,opt} = \inf_{(f_b,g)\in\Pi} \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{(f_b,g)}\left[\sum_{t=0}^{T-1} p(\Delta(t))\right], \tag{3.4}$$

where $f_b = (b, b, \ldots)$ represents an invariant buffer position assignment policy and $\bar{p}_{b,opt}$ is the optimal objective value in (3.4). The insights gained from solving this simplified problem (3.4) will subsequently guide us in deriving the optimal solution to the original scheduling problem (3.3).

Theorem 3.1 *If $|p(\delta)| \leq M$ for all δ and the T_i's are i.i.d. with $1 \leq \mathbb{E}[T_i] < \infty$, then $g = (S_1(\beta_b), S_2(\beta_b), \ldots)$ is an optimal solution to* (3.4)*, where*

$$S_{i+1}(\beta_b) = \min_{t \in \mathbb{Z}} \big\{ t \geq D_i(\beta_b) : \gamma(\Delta(t)) \geq \beta_b \big\}, \tag{3.5}$$

$D_i(\beta_b) = S_i(\beta_b) + T_i$ is the delivery time of the i-th feature submitted to the channel, $\Delta(t) = t - S_i(\beta_b) + b$ is the AoI at time t, $\gamma(\delta)$ is an index function, defined by

$$\gamma(\delta) = \inf_{\tau \in \{1,2,\ldots\}} \frac{1}{\tau} \sum_{k=0}^{\tau-1} \mathbb{E}\left[p(\delta + k + T_1) \right], \tag{3.6}$$

and the threshold β_b is the unique root of

$$\mathbb{E}\left[\sum_{t=D_i(\beta_b)}^{D_{i+1}(\beta_b)-1} p\big(\Delta(t)\big) \right] - \beta_b \, \mathbb{E}\left[D_{i+1}(\beta_b) - D_i(\beta_b) \right] = 0. \tag{3.7}$$

The optimal objective value to (3.4) *is given by*

$$\bar{p}_{b,opt} = \frac{\mathbb{E}\left[\sum_{t=D_i(\beta_b)}^{D_{i+1}(\beta_b)-1} p\big(\Delta(t)\big) \right]}{\mathbb{E}\left[D_{i+1}(\beta_b) - D_i(\beta_b) \right]}. \tag{3.8}$$

Furthermore, β_b is equal to the optimal objective value to (3.4)*, i.e., $\beta_b = \bar{p}_{b,opt}$.*

Proof 3.1 (Proof sketch) *The scheduling problem* (3.4) *is an infinite-horizon average-cost semi-Markov decision process (SMDP) [69, Chapter 5.6]. Define $\tau = S_{i+1} - D_i$ as the waiting time for sending the $(i+1)$-th feature after the i-th feature is delivered. The Bellman optimality equation of the SMDP* (3.4) *is*

$$\begin{aligned} h_b(\delta) = \inf_{\tau \in \{0,1,2,\ldots\}} \; & \mathbb{E}\left[\sum_{k=0}^{\tau+T_i-1} \left(p(\delta + k) - \bar{p}_{b,opt} \right) \right] \\ & + \mathbb{E}[h_b(T_i + b)], \;\; \delta = 1, 2, \ldots, \end{aligned} \tag{3.9}$$

where $h_b(\cdot)$ is the relative value function of the SMDP (3.4)*. Theorem 3.1 is proven by directly solving* (3.9)*. The details are provided in Appendix 3.A.*

In supervised learning algorithms, features are shifted, rescaled, and clipped during data pre-processing. Because of these pre-processing techniques, the inference error $p(\delta)$ is bounded. Therefore, the assumption $|p(\delta)| \leq M$ for all δ in Theorem 3.1 is not restrictive in practice.

The optimal scheduling policy in Theorem 3.1 is a threshold policy described by the index function $\gamma(\delta)$: According to (3.5), a feature is transmitted in time-slot t if and only if two conditions are satisfied: (i) The channel is available for scheduling in time-slot t and (ii) the index $\gamma(\Delta(t))$ exceeds a threshold β_b, which is precisely equal to the optimal objective value $\bar{p}_{b,opt}$ of (3.4). The expression of $\gamma(\delta)$ in (3.6) is obtained by solving the Bellman optimality equation (3.9), as explained in Appendix 3.A. The threshold β_b is calculated by solving the unique root of (3.7). Three low-complexity algorithms for this purpose were given by [25, Algorithms 1-3].

It is crucial to note that a non-monotonic AoI function $p(\delta)$ often leads to a non-monotonic index function $\gamma(\delta)$. Consequently, the inverse function of $\gamma(\delta)$ may not exist and the inequality $\gamma(\Delta(t)) \geq \beta_b$ in the threshold policy (3.5) cannot be equivalently rewritten as an inequality of the form $\Delta(t) \geq \alpha$. This distinction represents a significant departure from previous studies for minimizing either the AoI $\Delta(t)$ or its non-decreasing functions, e.g., [20–32]. In these earlier works, the solutions were usually expressed as threshold policies in the form $\Delta(t) \geq \alpha$. Our pursuit of a simple threshold policy for minimizing general and potentially non-monotonic AoI functions was inspired by the restart-in-state formulation of the Gittins index [70, Chapter 2.6.4], [71].

Now we present an optimal solution to (3.3).

Theorem 3.2 *If the conditions of Theorem 3.1 hold, then an optimal solution* (f^*, g^*) *to* (3.3) *is determined by*

(a) $f^* = (b^*, b^*, \ldots)$, *where*

$$b^* = \arg \min_{b \in \{0,1,\ldots,B-1\}} \bar{p}_{b,opt}, \tag{3.10}$$

and $\bar{p}_{b,opt}$ *is the optimal objective value to* (3.4).

(b) $g^* = (S_1^*, S_2^*, \ldots)$ *, where*

$$S_{i+1}^* = \min_{t \in \mathbb{Z}} \left\{ t \geq D_i^* : \gamma(\Delta(t)) \geq \bar{p}_{opt} \right\}, \tag{3.11}$$

$D_i^* = S_i^* + T_i$*,* $\gamma(\delta)$ *is defined in* (3.6)*, and* $\bar{p}_{opt}$ *is the optimal objective value of* (3.3)*, given by*

$$\bar{p}_{opt} = \min_{b \in \{0,1,\ldots,B-1\}} \bar{p}_{b,opt}. \tag{3.12}$$

Proof 3.2 *See Appendix 3.A.*

Theorem 3.2 suggests that, in the optimal solution to (3.3), one should select features from a fixed buffer position $b_i = b^*$. In addition, a feature is transmitted in time-slot t if and only if two conditions are satisfied: (i) The channel is available for transmission in time-slot t, (ii) the index $\gamma(\Delta(t))$ exceeds a threshold $\bar{p}_{opt}$ (i.e., $\gamma(\Delta(t)) \geq \bar{p}_{opt}$), where the threshold $\bar{p}_{opt}$ is exactly the optimal objective value of (3.3).

In the special case of a non-decreasing AoI function $p(\delta)$, it can be shown that the index function $\gamma(\delta) = \mathbb{E}[p(\delta + T_1)]$ is non-decreasing and $b^* = 0$ is the optimal buffer position in (3.10). The optimal strategy in such cases is to consistently select the freshest feature from the buffer such that $b_i = 0$. Hence, both the "generate-at-will" and "selection-from-buffer" models achieve the same minimum inference error. Furthermore, Theorem 3 in [23] can be directly derived from Theorem 3.2.

3.4 Data Driven Evaluations

In this section, we illustrate the performance of our scheduling policies, where we plug in the inference error versus AoI cost functions from the data driven experiments in Section 2.3. Then, we simulate the performance of different scheduling policies.

1. Generate-at-will, zero wait: The $(i+1)$-th feature sending time S_{i+1} is given by $S_{i+1} = D_i = S_i + T_i$ and the feature selection policy is $f = (0, 0, \ldots)$, i.e., $b_i = 0$ for all i.

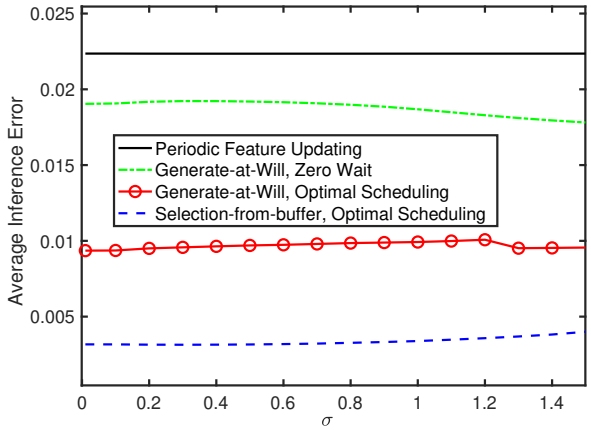


Figure 3.2: Time average inference error vs. the scale parameter σ of discretized i.i.d. log-normal transmission time distribution for single-source scheduling (in robot state prediction task).

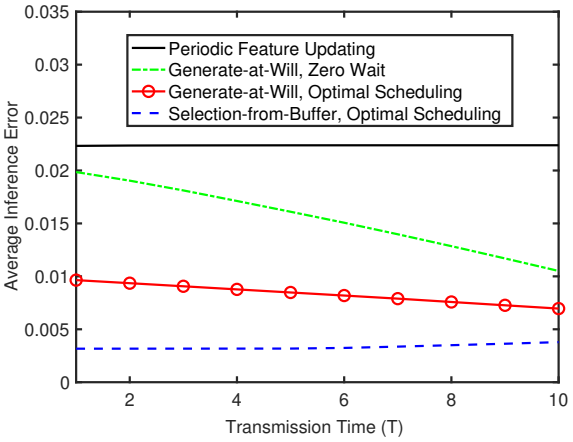


Figure 3.3: Time-average inference error vs. constant transmission time (T) (in robot state prediction task).

2. Generate-at-will, optimal scheduling: The policy is given by Theorem 3.1 with $b_i = 0$ for all i.

3. Selection-from-buffer, optimal scheduling: The policy is given by Theorem 3.2.

4. Periodic feature updating: Features are generated periodically with a period T_p and appended to a queue with buffer size B. When the buffer is full, no new feature is admitted to the buffer. Features in the buffer are sent over the channel in a first-come, first-served order.

Figs. 3.2-3.3 compare the time-averaged inference error of the four single-source scheduling policies defined earlier. These policies are evaluated using the inference error function, $p(\delta)$, obtained from the robot state prediction experiment of the leader-follower robotic system presented in Section 2.3 and illustrated in Fig. 2.2(c). The feature sequence length for this experiment is $u = 1$.

Fig. 3.2 illustrates the time-average inference error achieved by the four single-source scheduling policies defined above. The inference error function $p(\delta)$ used in this evaluation is illustrated in Fig. 2.2(c), which is generated by using the leader-follower robotic dataset and the trained neural network as explained in Chapter 2.3. The i-th feature transmission time T_i is assumed to follow a discretized i.i.d. log-normal distribution. In particular, T_i can be expressed as $T_i = \lceil \alpha e^{\sigma Z_i} / \mathbb{E}[e^{\sigma Z_i}] \rceil$, where Z_i's are i.i.d. Gaussian random variables with zero mean and unit variance. In Fig. 3.2, we plot the time average inference error versus the scale parameter σ of discretized i.i.d. log-normal distribution, where $\alpha = 1.2$, the buffer size is $B = 30$, and the period of uniform sampling is $T_p = 3$. The randomness of the transmission time increases with the growth of σ. Data-driven evaluations in Fig. 3.2 show that "selection-from-buffer" with optimal scheduler achieves 3 times performance gain compared to "generate-at-will," and 8 times performance gain compared to periodic feature updating.

Fig. 3.3 illustrates the performance of the four scheduling policies versus constant transmission time T. Similar to Fig. 3.2, the inference error function $p(\delta)$ is measured from leader-follower robotic dataset. This figure also shows that "selection-from-buffer" with optimal scheduler can achieve 8 time performance gain compared to periodic feature updating.

3.5 Conclusions

In this chapter, we design transmission scheduling policies for minimizing the inference error. Because fresher data is not always better, we propose a new "selection-from-buffer" status updating model. If the inference error is a non-decreasing function of the AoI, then the "selection-from-buffer" model achieves the same performance as the "generate-at-will"

model; if the AoI function is non-monotonic, the "selection-from-buffer" model can potentially achieve better performance. We devised an optimal "Selection-from-buffer" scheduling policy. The optimal scheduling policy can be expressed an *index-based threshold policy*, where a new packet is sent out whenever the index $\gamma(\Delta(t))$ exceeds a pre-determined threshold. The threshold can be computed by using low complexity algorithms. The results in this chapter hold (i) for minimizing general AoI functions (monotonic or non-monotonic) and (ii) for random delay channels. Data-driven evaluations show that the optimal scheduler has the potential to achieve up to 3 times smaller inference error compared to "generate-at-will" with optimal scheduling strategy and 8 times smaller inference error compared to periodic feature updating.

Appendix

3.A Proof of Theorem 3.2

In this section, we prove Theorem 3.1 and Theorem 3.2. These theorems provide optimal solutions for the scheduling problems (3.3) and (3.4). We begin by deriving the optimal solution for (3.3). Subsequently, the optimal solutions for (3.4), follow directly, as these problem is a special case of (3.3).

We rewrite the problem (3.3) as follows:

$$\bar{p}_{opt} = \inf_{\pi \in \Pi} \limsup_{T \to \infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} p(\Delta(t))\right], \tag{3.13}$$

where $p(\Delta(t))$ is the penalty at time t, $\Delta(t) \in \mathbb{Z}^+$ is the AoI, $\pi = ((S_1, b_1), (S_2, b_2), \ldots)$ is a scheduling policy, Π is the set of all causal and signal-agnostic scheduling policies, and $\bar{p}_{opt}$ is the optimal objective value to (3.3).

The scheduling problem (3.3) is an infinite-horizon average-cost semi-Markov decision process (SMDP) [69, Chapter 5.6]. We provide a detailed description of the components of this problem:

- **Decision Time:** Each i-th feature delivery time $D_i = S_i + T_i$ is a decision time of the problem (3.13), where S_i is the scheduling time of the i-th feature and the i-th feature takes $T_i \geq 1$ time slots to be delivered.

- **State:** At time slot D_i, the state of the system is represented by AoI $\Delta(D_i)$.

- **Action:** Let $\tau_{i+1} = S_{i+1} - D_i$ represent the waiting time for sending the $(i+1)$-th feature after the i-th feature is delivered. As we consider $S_0 = 0$ and $S_i = \sum_{j=1}^{i}(T_{j-1} + \tau_j)$ for each $i = 1, 2, \ldots$. Given $(T_0, T_1, \ldots)$, the sequence $(S_1, S_2, \ldots)$ is uniquely determined by $(\tau_1, \tau_2, \ldots)$. Hence, one can also use $(\tau_1, \tau_2, \ldots)$ to represent a sequence of actions instead of $(S_1, S_2, \ldots)$.

  At time D_i, the scheduler decides the waiting time τ_{i+1} and the buffer position b_{i+1}.

- **State Transitions:** The AoI process $\Delta(t)$ evolves as

$$\Delta(t) = \begin{cases} T_i + b_i, & \text{if } t = D_i, i = 0, 1, \ldots, \\ \Delta(t-1) + 1, & \text{otherwise.} \end{cases} \tag{3.14}$$

- **Expected Transition Time:** The expected time difference between two decision times, D_i and D_{i+1}, is given by

$$\begin{aligned} \mathbb{E}[D_{i+1} - D_i] &= \mathbb{E}[S_{i+1} + T_{i+1} - (S_i + T_i)] \\ &= \mathbb{E}[S_i + T_i + \tau_{i+1} + T_{i+1} - S_i - T_i] \\ &= \mathbb{E}[\tau_{i+1} + T_{i+1}]. \end{aligned} \tag{3.15}$$

- **Expected Transition Cost:** The expected cumulative cost incurred during the interval between two decision times, D_i and D_{i+1}, can be calculated as

$$\mathbb{E}\left[\sum_{t=D_i}^{D_{i+1}-1} \Big(p(\Delta(t)) + \lambda c(t) \Big)\right] = \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} p(\Delta(D_i + k))\right]. \tag{3.16}$$

The infinite-horizon average-cost SMDP (3.3) can be solved by using dynamic programming [69, 72]. Let $h : \mathbb{Z}^+ \mapsto \mathbb{R}$ be the relative value function associated with the average-cost SMDP (3.3). At time $t = D_i$, the optimal action (τ_{i+1}, b_{i+1}) can be determined by solving the following Bellman optimality equation [69, P. 275]:

$$\begin{aligned} h(\Delta(D_i)) &= \inf_{\substack{\tau_{i+1} \in \{0,1,\ldots\} \\ b_{i+1} \in \{0,\ldots,B-1\}}} \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} p(\Delta(D_i + k))\right] - \bar{p}_{opt}\mathbb{E}[\tau_{i+1} + T_{i+1}] + \mathbb{E}[h(\Delta(D_{i+1})] \\ &= \inf_{\substack{\tau_{i+1} \in \{0,1,\ldots\} \\ b_{i+1} \in \{0,\ldots,B-1\}}} \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} \Big(p(\Delta(D_i + k)) - \bar{p}_{opt} \Big)\right] + \mathbb{E}[h(\Delta(D_{i+1})] \\ &= \inf_{\substack{\tau_{i+1} \in \{0,1,\ldots\} \\ b_{i+1} \in \{0,\ldots,B-1\}}} \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} \Big(p(\Delta(D_i + k)) - \bar{p}_{opt} \Big)\right] + \mathbb{E}[h(T_{i+1} + b_{i+1})], \end{aligned} \tag{3.17}$$

where the last equality holds because $\Delta(D_{i+1}) = T_{i+1} + b_{i+1}$.

From (3.17), it is observed that the buffer position b_{i+1} only depends on the term $\mathbb{E}[h(T_{i+1}+b_{i+1})]$, while the waiting time τ_{i+1} has no impact on $\mathbb{E}[h(T_{i+1}+b_{i+1})]$. Hence, the optimal buffer position b_{i+1}^* is determined by

$$b_{i+1}^* = \arg \min_{b_{i+1}\in\{0,1,\ldots,B-1\}} \mathbb{E}[h(T_{i+1}+b_{i+1})]. \tag{3.18}$$

Since T_i's are i.i.d., $\mathbb{E}[h(T_{i+1}+b)] = \mathbb{E}[h(T_i+b)] = \cdots = \mathbb{E}[h(T_1+b)]$ for all i and b. Hence, from (3.18), it is evident that there exists a $b^* \in \{0,1,\ldots,B-1\}$ such that $b_1^* = b_2^* = \cdots = b_{i+1}^* = b^*$ that satisfies

$$b^* = \arg \min_{b\in\{0,1,\ldots,B-1\}} \mathbb{E}[h(T_1+b)]. \tag{3.19}$$

Because the optimal buffer position is time-invariant, the problem (3.13) can be expressed as

$$\bar{p}_{opt} = \min_{b\in\{0,1,\ldots,B-1\}} \bar{p}_{b,opt}, \tag{3.20}$$

where $\bar{p}_{b,opt}$ is given by

$$\bar{p}_{b,opt} = \inf_{\pi_b\in\Pi_b} \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\pi_b}\left[\sum_{t=0}^{T-1} p(\Delta(t))\right], \tag{3.21}$$

$\pi_b = ((S_1,b),(S_2,b),\ldots)$, Π_b is the set of all causal and signal-agnostic scheduling policies π_b with fixed buffer position b, and $\bar{p}_{b,opt}$ is the optimal objective value to (3.21).

At every i-th decision time D_i of the average-cost SMDP (3.21), the scheduler decides the waiting time τ_{i+1}. The average-cost SMDP (3.21) can be solved by using dynamic programming [69,72]. Given AoI value δ at decision time D_i, the Bellman optimality equation of (3.21) is obtained by substituting $\Delta(D_i) = \delta$, $b_{i+1} = b$, and $\bar{p}_{opt} = \bar{p}_{b,opt}$ into (3.17), given

by

$$h_b(\delta) = \inf_{\tau\in\{0,1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_{i+1}-1} (p(\delta+k) - \bar{p}_{b,opt})\right] + \mathbb{E}[h_b(T_{i+1}+b)],\ \delta = 1,2,\ldots$$
$$= \inf_{\tau\in\{0,1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} (p(\delta+k) - \bar{p}_{b,opt})\right] + \mathbb{E}[h_b(T_1+b)],\ \delta = 1,2,\ldots, \qquad (3.22)$$

where the last equality holds because T_i's are identically distributed. Let $\tau(\delta, \bar{p}_{b,opt})$ be an optimal solution to (3.22). If $\Delta(D_i) = \delta$, then an optimal waiting time τ_{i+1} of (3.21) for sending the $(i+1)$-th feature is $\tau(\delta, \bar{p}_{b,opt})$.

From (3.22), we can show that $\tau(\delta, \bar{p}_{b,opt}) = 0$ if

$$\inf_{\tau\in\{1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} (p(\delta+k) - \bar{p}_{b,opt})\right] \geq \mathbb{E}\left[\sum_{k=0}^{T_1-1} (p(\delta+k) - \bar{p}_{b,opt})\right]. \qquad (3.23)$$

After some rearrangement, the inequality (3.23) can also be expressed as

$$\inf_{\tau\in\{1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau-1} (p(\delta+k+T_1) - \bar{p}_{b,opt})\right] \geq 0. \qquad (3.24)$$

Next, similar to [25, Lemma 7], the following lemma holds.

Lemma 3.1 *The inequality* (3.24) *holds if and only if*

$$\inf_{\tau\in\{1,2,\ldots\}} \frac{1}{\tau}\mathbb{E}\left[\sum_{k=0}^{\tau-1} p(\delta+k+T_1)\right] \geq \bar{p}_{b,opt}. \qquad (3.25)$$

According to (3.6), the left-hand side of (3.25) is equal to $\gamma(\delta)$. Similarly, $\tau(\delta, \bar{p}_{b,opt}) = 1$, if $\tau(\delta, \bar{p}_{b,opt}) \neq 0$ and

$$\inf_{\tau\in\{2,3,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau-1} (p(\delta+k+T_1) - \bar{p}_{b,opt}\right] \geq 0 \qquad (3.26)$$

By using Lemma 3.1 and (3.6), we can show that the inequality (3.26) can be rewritten as

$$\gamma(\delta+1) \geq \bar{p}_{b,opt}. \qquad (3.27)$$

By repeating this process, we get $\tau(\delta, \bar{p}_{b,opt}) = k$ is optimal, if $\tau(\delta, \bar{p}_{b,opt}) \neq 0, 1, \ldots, k-1$ and

$$\gamma(\delta + k) \geq \bar{p}_{b,opt}. \tag{3.28}$$

Hence, the optimal waiting time $\tau_{i+1} = \tau(\delta, \bar{p}_{b,opt})$ is determined by

$$\tau(\delta, \bar{p}_{b,opt}) = \min_{k \in \mathbb{Z}} \{k \geq 0 : \gamma(\delta + k) \geq \bar{p}_{b,opt}\}. \tag{3.29}$$

Now, we are ready to compute the optimal objective value $\bar{p}_{b,opt}$. Using (3.22), we can determine the value of $\mathbb{E}[h_b(T_1 + b)]$, which is given by

$$\mathbb{E}[h_b(T_1 + b)] = \mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\bar{p}_{b,opt})+T_1-1} (p(T_1 + b + k) - \bar{p}_{b,opt})\right] + \mathbb{E}[h_b(T_1 + b)], \tag{3.30}$$

which yields

$$\mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\bar{p}_{b,opt})+T_1-1} \Big(p(T_1 + b + k) - \bar{p}_{b,opt}\Big)\right] = 0. \tag{3.31}$$

Rearranging (3.31), we get

$$\begin{aligned} \bar{p}_{b,opt} &= \frac{\mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\bar{p}_{b,opt})+T_1-1} p(T_1 + b + k)\right]}{\mathbb{E}[\tau(T_1 + b, \bar{p}_{b,opt}) + T_1]} \\ &= \frac{\mathbb{E}\left[\sum_{t=D_i(\bar{p}_{b,opt})}^{D_{i+1}(\bar{p}_{b,opt})-1} p\big(\Delta(t)\big)\right]}{\mathbb{E}\left[D_{i+1}(\bar{p}_{b,opt}) - D_i(\bar{p}_{b,opt})\right]}, \end{aligned} \tag{3.32}$$

where $D_{i+1}(\bar{p}_{b,opt}) = S_{i+1}(\bar{p}_{b,opt}) + T_{i+1}$ and

$$S_{i+1}(\bar{p}_{b,opt}) = \min_{t \geq 0} \{t \geq D_i(\bar{p}_{b,opt}) : \gamma(\Delta(t)) \geq \bar{p}_{b,opt}\}, \tag{3.33}$$

and the last equality holds due to (3.15) and (3.16).

Now, by combining (3.20), (3.29), and (3.32), we obtain optimal solution to (3.4) and (3.3).

Finally, we need to prove that

$$\mathbb{E}\left[\sum_{t=D_i(\beta)}^{D_{i+1}(\beta)-1} p\big(\Delta(t)\big)\right] - \beta\mathbb{E}\left[D_{i+1}(\beta) - D_i(\beta)\right] = 0. \tag{3.34}$$

has a unique root. We define

$$J(\beta) = \mathbb{E}\left[\sum_{t=D_i(\beta)}^{D_{i+1}(\beta)-1} p\big(\Delta(t)\big)\right] - \beta\mathbb{E}\left[D_{i+1}(\beta) - D_i(\beta)\right]. \tag{3.35}$$

Lemma 3.2 *The function $J(\beta)$ has the following properties:*

(i) The function $J(\beta)$ is concave, continuous, and strictly decreasing in β.

(ii) $\lim_{\beta\to\infty} j(\beta) = -\infty$ *and* $\lim_{\beta\to-\infty} j(\beta) = \infty$.

Proof 3.3 *See Appendix 3.B*

Lemma 3.2 proves the uniqueness of (3.37). Also, the uniqueness of the root of (3.7) follows immediately from Lemma 3.2.

3.B Proof of Lemma 3.2

We denote $D_{i+1}(\beta) = S_{i+1}(\beta) + T_{i+1}$, $S_{i+1}(\beta) = \tau(\Delta(D_i), \beta) + D_i(\beta)$, where $\tau(\delta, \beta)$ is defined as the optimal solution of

$$\inf_{\tau\in\{0,1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} \left(p(\delta+k) - \beta\right)\right]. \tag{3.36}$$

Because T_i's are i.i.d., and $\Delta(D_i) = T_i + b$, we can express (3.35) as

$$\begin{aligned} J(\beta) &= \mathbb{E}\left[\sum_{t=D_i(\beta)}^{D_{i+1}(\beta)-1} p\big(\Delta(t)\big)\right] - \beta\mathbb{E}\left[D_{i+1}(\beta) - D_i(\beta)\right] \\ &= \mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\beta)+T_1-1} \left(p(T_1+b+k) - \beta\right)\right] \\ &= \inf_{\tau\in\{0,1,\ldots} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} \left(wp(T_1+b+k) - \beta\right)\right]. \end{aligned} \tag{3.37}$$

Since the right-hand side of (3.37) is the pointwise infimum of the linear decreasing functions of β, $J(\beta)$ is concave, continuous, and strictly decreasing in β. This completes the proof of part (i) of Lemma 3.2.

Part (ii) of Lemma 3.2 holds because for any $\tau \geq 0$

$$\lim_{\beta\to\infty} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} \left(p(T_1+b+k) - \beta\right)\right] = -\infty \tag{3.38}$$

and

$$\lim_{\beta\to-\infty} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} \left(p(T_1+b+k) - \beta\right)\right] = \infty. \tag{3.39}$$

This completes the proof.

Chapter 4

Timely Inference in Multi-Source and Multi-Channel Networks

4.1 Introduction

In many networked intelligent systems, a receiver (e.g., an autonomous vehicle) requires information from multiple sources (e.g., onboard cameras, roadside units, and nearby vehicles). Due to limited communication resources, it is impractical to obtain information from all sources simultaneously. A scheduler decides which sources to select and what information from sources to send to accomplish a task. Due to resource constraints, the efficient design of multi-source scheduling policy is pivotal to guarantee low-latency and reliable performance of networked intelligent systems. Towards that end, in this chapter, we develop a novel transmission scheduling policy to minimize the weighted summation inference error in multi-source, multi-channel remote inference systems. The contributions of this chapter are summarized as follows

- When there are multiple source-predictor pairs and multiple channels, the scheduling problem is a restless multi-armed bandit (RMAB) problem with multiple actions. We propose a multi-source, multi-action scheduling policy that uses a Whittle index algorithm to determine which sources to schedule and employs a duality-based selection-from-buffer algorithm to decide which features to schedule from the buffers of these sources. By utilizing linear programming (LP)-based priority conditions [73, 74], we establish the asymptotic optimality of this scheduling policy as the numbers of sources and channels tend to infinity, maintaining a constant ratio (see Theorem 4.6).

- The above results hold (i) for minimizing general AoI functions (monotonic or non-monotonic) and (ii) for random delay channels. Numerical results validate the asymptotic optimality of the proposed scheduling policy (see Figs. 4.2-4.3).

- When the training and inference data have the same probabilistic distribution, multi-source multi-channel remote inference reduces to signal-agnostic multi-source multi-channel remote estimation. Hence, the results of the present work above also apply to signal-agnostic multi-source multi-channel remote estimation.

4.1.1 Related Works

The optimization of linear and non-linear functions of AoI for multi-source scheduling is a restless multi-armed bandit (RMAB) problem. The multi-source problems in previous AoI studies [26, 29–32, 39] focused only on source selection, which is an RMAB problem with binary actions determining whether to select a source or not. Moreover, the previous AoI studies considered monotonic AoI functions. Whittle index policy [33] was used to solve the RMAB problems with binary actions and monotonic AoI penalty functions [26, 29–32, 39]. Our multi-source problem is an RMAB with multiple actions and possibly a non-monotonic AoI penalty function. Because of the multiple-action setup, the Whittle index alone can not be utilized to solve our problem. Consequently, we design a new asymptotically optimal policy for multi-action RMAB with general AoI functions (monotonic or non-monotonic).

This paper is also related to the field of signal-agnostic remote estimation. The prior studies [23, 25, 27, 39, 40, 44, 45] in signal-agnostic remote estimation focused on Gaussian and Markovian processes. The results presented in the current paper are applicable to more general processes.

4.2 Multi-Source, Multi-Channel Status Updating Model

Consider the remote inference system depicted in Fig. 4.1, which consists of M source-predictor pairs and N channels. Each source adopts a "selection-from-buffer" model: At the beginning of time slot t, each source m generates a feature $X_{m,t}$ and adds it into the buffer that stores B_m most recent features $(X_{m,t}, \ldots, X_{m,t-B_m+1})$, meanwhile the oldest feature $X_{m,t-B_m}$ is removed from the buffer. At each time slot t, a central scheduler decides: (i) which sources to select and (ii) which features from the buffer of selected sources to send. Each feature transmission lasts for one or multiple time slots. We consider non-preemptive

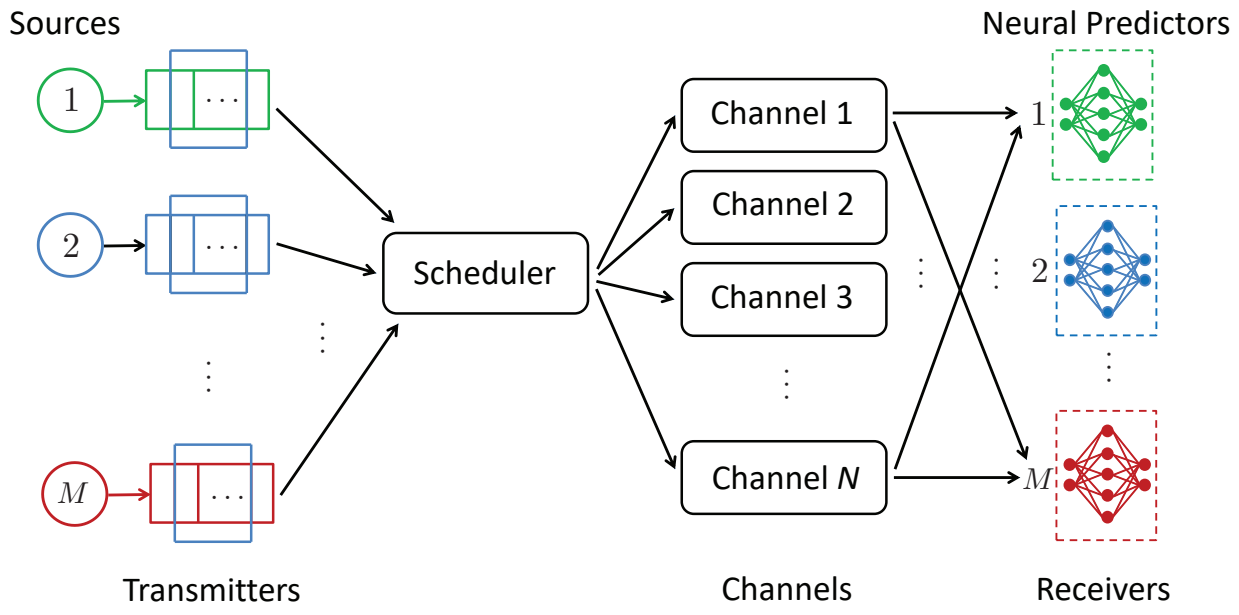


Figure 4.1: A multi-source, multi-channel remote inference system.

transmissions, i.e., once a channel starts to send a feature, the channel must finish serving that feature before switching to serve another feature. At any given time slot, each source can be served by no more than one channel. We use an indicator variable $c_m(t) \in \{0, 1\}$ to represent whether a feature from source m occupies a channel at time slot t, where $c_m(t) = 1$ if source m is being served by a channel at time slot t; otherwise, $c_m(t) = 0$. Once a feature is delivered, an acknowledgment is fed back to the scheduler within the same time-slot. By this, the channel occupation status $c_m(t)$ is known to the scheduler at every time slot t. Due to limited channel resources, the system must satisfy the constraint $\sum_{m=1}^{M} c_m(t) \leq N$ for all time slot $t = 0, 1, \ldots$.

The system starts to operate at time slot $t = 0$. The i-th feature sent by source m is generated in time slot $G_{m,i}$, submitted to a channel in time slot $S_{m,i}$, and delivered to the receiver in time slot $D_{m,i} = S_{m,i} + T_{m,i}$, where $G_{m,i} \leq S_{m,i} < D_{m,i}, D_{m,i} \leq S_{m,i+1} < D_{m,i+1}$, and $T_{m,i} \geq 1$ is the feature transmission time of the i-th feature sent from source m. We assume that the $T_{m,i}$'s are independent across the sources and are i.i.d. for features originating from the same source with $1 \leq \mathbb{E}[T_{m,i}] < \infty$.

Scheduling Policies and Problem Formulation

In time slot $S_{m,i}$, let $X_{G_{m,i}} = X_{S_{m,i}-b_{m,i}}$ be the feature submitted to a channel from source m, which is the $(b_{m,i}+1)$-th freshest feature in source m's buffer, with $b_{m,i} \in \{0, 1, \ldots, B_m - 1\}$. By this, a scheduling policy for source m is denoted by (f_m, g_m), where $g_m = (S_{m,1}, S_{m,2}, \ldots)$ determines when to schedule source m, and $f_m = (b_{m,1}, b_{m,2}, \ldots)$ specifies which feature to send from source m's buffer.

Let $U_m(t) = \max_i\{G_{m,i} : D_{m,i} \leq t\}$ represent the generation time of the freshest feature delivered from source m to the receiver up to time slot t. Because $G_{m,i} = S_{m,i} - b_{m,i}$, $U_m(t) = \max_i\{S_{m,i} - b_{m,i} : D_{m,i} \leq t\}$. The age of information (AoI) of source m at time slot t is

$$\Delta_m(t) = t - U_m(t) = t - \max_i\{S_{m,i} - b_{m,i} : D_{m,i} \leq t\}. \tag{4.1}$$

The initial state of the system is assumed to be $S_{m,0} = 0, D_{m,0} = T_{m,0}$, and $\Delta_m(0)$ is a finite constant.

Let Π_m denote the set of all causal and signal-agnostic scheduling policies (f_m, g_m) that satisfy the following conditions: (i) the transmission time schedule $S_{m,i}$ and the buffer position $b_{m,i}$ are determined based on the current and the historical information available at the scheduler; (ii) source m can be served by at most one channel at a time and feature transmissions are non-preemptive; (iii) the scheduler does not have access to the realization of the feature and the target processes; and (iv) the scheduler can access the inference error function $\text{err}_{\text{inference},m}(\Delta_m(t))$ and the distribution of $T_{m,i}$ for source m. We define Π as the set of all causal and signal-agnostic scheduling policies $\pi = (f_m, g_m)_{m=1}^M$.

Our goal is to find a scheduling policy that minimizes the weighted summation of the time-averaged expected inference errors of the M sources:

$$\inf_{\pi\in\Pi} \limsup_{T\to\infty} \sum_{m=1}^{M} w_m \, \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} p_m(\Delta_m(t))\right], \tag{4.2}$$

$$\text{s.t.} \sum_{m=1}^{M} c_m(t) \leq N, t = 0, 1, \ldots, \tag{4.3}$$

$$c_m(t) \in \{0, 1\}, m = 1, 2, \ldots, M, t = 0, 1, \ldots, \tag{4.4}$$

where $p_m(\Delta_m(t)) = \text{err}_{\text{inference},m}(\Delta_m(t))$ is the inference error of source m at time slot t and $w_m > 0$ is the weight of source m.

Let $d_m(t) \in \{0, 1, \ldots\}$ denote the amount of time that has been spent to send the current feature of source m by time slot t. Hence, $d_m(t) = 0$ if no feature of source m is in service at time slot t, and $d_m(t) > 0$ if a feature of source m is currently in service at time slot t. Problem (4.2)-(4.4) is a multi-action Restless Multi-armed Bandit (RMAB) problem, in which $(\Delta_m(t), d_m(t))$ is the state of the m-th bandit. At time slot t, if a feature from source m is submitted to a channel, bandit m is said to be *active*; otherwise, if source m is not under service or if one feature of source m is under service whereas the service started before time slot t, then bandit m is said to be *passive*. The bandits are "restless" because the state $(\Delta_m(t), d_m(t))$ undergoes changes even when the m-th bandit is passive [26, 33]. When a bandit m is activated, the scheduler can select any of the B_m features from the buffer of source m. Thus, this problem is a multi-action RMAB.

4.3 An Asymptotically Optimal Scheduling Solution

It is well-known that RMAB with binary actions is PSPACE-hard [75]. RMABs with multiple actions, like (4.2)-(4.4), would be even more challenging to solve. In the sequence, we will generalize the conventional Whittle index theoretical framework [33] for binary-action RMABs, by developing a new index-based scheduling policy and proving this policy is asymptotically optimal for solving the multi-action RMAB problem (4.2)-(4.4). This new theoretical framework contains four steps: (a) We first reformulate (4.2)-(4.4) as an equivalent multi-action RMAB problem with an equality constraint by using dummy bandits [39,73]. The usage of dummy bandits is necessary for establishing the asymptotic optimality result in subsequent steps. (b) We relax the per-time-slot channel constraint as a time-average expected channel constraint, solve the relaxed problem by using Lagrangian dual optimization, and compute the optimal dual variable λ^*. (c) Problem (4.2)-(4.4) requires to determine (i) which source to schedule and (ii) which feature from the buffer of the scheduled source to send. In the proposed scheduling policy, the former is decided by a Whittle index policy, for which we establish indexability and derive an analytical expression of the Whittle index. The latter is determined by a λ^*-based selection-from-buffer policy. (d) Finally, we

employ LP priority-based sufficient condition [73, 74] to prove that the proposed policy is asymptotically optimal as the numbers of users and channels increase to infinite with a fixed ratio.

Dummy Bandits and Constraint Relaxation

Besides the original M bandits, we introduce N additional *dummy bandits* (the number of dummy bandits needs to be equal to the number of channels) that satisfy two conditions: (i) each dummy bandit has a zero age penalty function $p_0(\Delta_0(t)) = 0$; (ii) when activated, each dummy bandit occupies a channel for one time slot. Let $c_0(t) \in \{0, 1, \ldots, N\}$ be the number of dummy bandits that are activated in time slot t. Let $\pi_0 = \{c_0(t), t = 0, 1, \ldots\}$ be a scheduling policy for the dummy bandits and Π_0 be the set of all policies π_0. Using these dummy bandits, (4.2)-(4.4) is reformulated as an RMAB with equality constraints (4.6), i.e.,

$$\inf_{\pi\in\Pi,\pi_0\in\Pi_0} \limsup_{T\to\infty} \sum_{m=1}^{M} w_m \, \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} p_m(\Delta_m(t))\right], \tag{4.5}$$

$$\text{s.t.} \sum_{m=0}^{M} c_m(t) = N, t = 0, 1, \ldots, \tag{4.6}$$

$$c_0(t) \in \{0, 1, \ldots, N\}, t = 0, 1, \ldots, \tag{4.7}$$

$$c_m(t) \in \{0, 1\}, m = 1, 2, \ldots, M, t = 0, 1, \ldots. \tag{4.8}$$

Because the number of dummy bandit is equal to the number of channels and when activated, each dummy bandit occupies a channel for one time slot, we always can have the equality constraint (4.6). Moreover, dummy bandits do not incur additional cost, the scheduling policy that optimizes (4.2)-(4.4) also optimizes (4.5)-(4.8).

Now, we replace the per-slot channel constraints (4.6) by a time-average expected channel constraint (4.10), which produces the following relaxed problem:

$$\inf_{\pi\in\Pi,\pi_0\in\Pi_0} \limsup_{T\to\infty} \sum_{m=1}^{M} w_m \, \mathbb{E}\left[\sum_{t=0}^{T-1} p_m(\Delta_m(t))\right], \tag{4.9}$$

$$\text{s.t. } \limsup_{T\to\infty} \sum_{m=0}^{M} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} c_m(t)\right] = N, \tag{4.10}$$

$$c_0(t) \in \{0, 1, \ldots, N\}, t = 0, 1, \ldots, \tag{4.11}$$

$$c_m(t) \in \{0, 1\}, m = 1, 2, \ldots, M, t = 0, 1, \ldots. \tag{4.12}$$

The optimal objective value of (4.9)-(4.12) provides a lower bound of the optimal objective value of (4.2)-(4.4).

Lagrangian Dual Optimization for Solving (4.9)**-**(4.12)

We solve the relaxed problem (4.9)-(4.12) by Lagrangian dual optimization [33, 76]. To that end, we associate a Lagrangian multiplier $\lambda \in \mathbb{R}$ to the constraint (4.10) and get the following dual function

$$q(\lambda) = \inf_{\pi\in\Pi,\pi_0\in\Pi_0} \limsup_{T\to\infty} \mathbb{E}\Bigg[\frac{1}{T}\sum_{t=0}^{T-1}\Bigg(\sum_{m=1}^{M} w_m p_m(\Delta_m(t)) + \lambda\Bigg(\sum_{m=0}^{M} c_m(t) - N\Bigg)\Bigg)\Bigg], \tag{4.13}$$

where $\lambda \in \mathbb{R}$ is also referred to as the transmission cost. The dual optimization problem is given by

$$\lambda^* = \arg\max_{\lambda\in\mathbb{R}} q(\lambda), \tag{4.14}$$

where λ^* is the optimal dual solution.

Solution to (4.13)

The problem (4.13) can be decomposed into $(M+1)$ sub-problems. For $m=0$, the sub-problem associated to the dummy bandits is given by

$$\inf_{\pi_0 \in \Pi_0} \limsup_{T\to\infty} \mathbb{E}_{\pi_0} \left[\frac{1}{T} \sum_{t=0}^{T-1} \lambda c_0(t) \right]. \tag{4.15}$$

Theorem 4.1 *If $\lambda > 0$, the optimal solution to* (4.15) *is $c_0^*(t) = 0$ for all t; if $\lambda \leq 0$, the optimal solution to* (4.15) *is $c_0^*(t) = N$ for all t.*

For each $m = 1, 2, \ldots, M$, the sub-problem associated with bandit m is given by

$$\bar{p}_{m,opt}(\lambda) = \inf_{(f_m, g_m) \in \Pi_m} \limsup_{T\to\infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} w_m p_m(\Delta_m(t)) + \lambda c_m(t) \right], \tag{4.16}$$

where $\bar{p}_{m,opt}(\lambda)$ is the optimal objective value to (4.16).

To explain the optimal solution to (4.16), we first fix the buffer position at $b_{m,i} = b$ for all i and optimize the transmission time schedule $g_m = (S_{m,1}, S_{m,2}, \ldots)$. This simplified problem is formulated as

$$\bar{p}_{m,b,opt}(\lambda) = \inf_{(f_{m,b}, g_m) \in \Pi_m} \limsup_{T\to\infty} \mathbb{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} w_m p_m(\Delta_m(t)) + \lambda c_m(t) \right], \tag{4.17}$$

where $f_{m,b} = (b, b, \ldots)$ and $\bar{p}_{m,b,opt}(\lambda)$ is the optimal objective value in (4.17).

Theorem 4.2 *If $T_{m,i}$'s are i.i.d. with $1 \leq \mathbb{E}[T_{m,i}] < \infty$, then there exists an optimal solution $g_m(\lambda) = (S_{m,1}(\beta_{m,b}(\lambda)), S_{m,2}(\beta_{m,b}(\lambda)), \ldots)$ to* (4.17)*, where*

$$S_{m,i+1}(\beta_{m,b}(\lambda)) = \min_{t \in \mathbb{Z}} \big\{ t \geq D_{m,i}(\beta_{m,b}(\lambda)) : \gamma_m(\Delta_m(t)) \geq \beta_{m,b}(\lambda) \big\}, \tag{4.18}$$

$D_{m,i}(\beta_{m,b}(\lambda)) = S_{m,i}(\beta_{m,b}(\lambda)) + T_{m,i}$ is the delivery time of the i-th feature submitted to the channel, $\Delta_m(t) = t - S_{m,i}(\beta_{m,b}(\lambda)) + b$ is the AoI at time t, $\gamma_m(\delta)$ is an index function,

defined by

$$\gamma_m(\delta) = \inf_{\tau \in \{1,2,\ldots\}} \frac{1}{\tau} \sum_{k=0}^{\tau-1} \mathbb{E}\left[w_m p_m(\delta + k + T_{m,1})\right], \tag{4.19}$$

and the threshold $\beta_{m,b}(\lambda)$ *is the unique root of*

$$\mathbb{E}\left[\sum_{t=D_{m,i}(\beta_{m,b}(\lambda))}^{D_{m,i+1}(\beta_{m,b}(\lambda))-1} w_m p_m\big(\Delta_m(t)\big)\right] + \lambda\ \mathbb{E}[T_{m,i}]$$
$$-\ \beta_{m,b}(\lambda)\ \mathbb{E}\left[D_{m,i+1}(\beta_{m,b}(\lambda)) - D_{m,i}(\beta_{m,b}(\lambda))\right] = 0. \tag{4.20}$$

Furthermore, $\beta_{m,b}(\lambda)$ *is equal to the optimal objective value to* (4.17)*, i.e.,* $\beta_{m,b}(\lambda) = \bar{p}_{m,b,opt}(\lambda)$.

Proof 4.1 *See Appendix 4.A.*

Now we present an optimal solution to (4.16).

Theorem 4.3 *If the conditions of Theorem 4.2 hold, then an optimal solution* $(f_m^*(\lambda), g_m^*(\lambda))$ *to* (4.16) *is determined by*

(a) $f_m^*(\lambda) = (b_m^*(\lambda), b_m^*(\lambda), \ldots)$*, where*

$$b_m^*(\lambda) = \arg \min_{b \in \{0,1,\ldots,B_m-1\}} \bar{p}_{m,b,opt}(\lambda), \tag{4.21}$$

and $\bar{p}_{m,b,opt}(\lambda)$ *is the optimal objective value to* (4.17)*.*

(b) $g_m^*(\lambda) = (S_{m,1}^*(\lambda), S_{m,2}^*(\lambda), \ldots)$*, where*

$$S_{m,i+1}^*(\lambda) = \min_{t \in \mathbb{Z}} \big\{t \geq D_{m,i}^* : \gamma_m(\Delta_m(t)) \geq \bar{p}_{m,opt}(\lambda)\big\}, \tag{4.22}$$

$D_{m,i}^*(\lambda) = S_{m,i}^*(\lambda) + T_{m,i}$*,* $\gamma_m(\delta)$ *is defined in* (4.19)*, and* $\bar{p}_{m,opt}(\lambda)$ *is the optimal objective value of* (4.16)*, given by*

$$\bar{p}_{m,opt}(\lambda) = \min_{b \in \{0,1,\ldots,B_m-1\}} \bar{p}_{m,b,opt}(\lambda). \tag{4.23}$$

Proof 4.2 *See Appendix 4.A.*

Solution to (4.14)

Next, we solve the dual problem (4.14). Let $c^*_{0,\lambda}(t) \in \{0, 1, \ldots, N\}$ be the number of dummy bandits activated in time slot t in the optimal solution to (4.15) and let $c^*_{m,\lambda}(t) \in \{0, 1\}$ denote whether source m is under service in time slot t in the optimal solution to (4.16). The dual problem (4.14) is solved by the following stochastic sub-gradient algorithm:

$$\lambda_{k+1} = \lambda_k + \frac{\alpha}{k}\left\{\frac{1}{T}\sum_{m=0}^{M}\sum_{t=0}^{T-1} c^*_{m,\lambda_k}(t) - N\right\}, \tag{4.24}$$

where $\alpha/k > 0$ is the step size and $T > 1$ is a sufficient large integer. In the k-th iteration, let $\lambda = \lambda_k$ and run the optimal solution to (4.13) for T time slots, then execute the dual update (4.24).

A Scheduling Policy for the Original Problem (4.2)-(4.4)

Now, we develop a scheduling policy for the original multi-action RMAB problem (4.2)-(4.4). The proposed policy contains two parts: (i) a Whittle index policy is used to determine which sources to schedule, and (ii) a λ^*-based selection-from-buffer policy is employed to determine which features to choose from the buffers of the scheduled sources.

Whittle Index-based Source Scheduling Policy

The Whittle index theory only applies to RMAB problems that are indexable [33]. Hence, we first establish the indexability of problem (4.5)-(4.8). Recall that $(\Delta_m(t), d_m(t))$ is the state of the m-th bandit, where $\Delta_m(t)$ is the AoI and $d_m(t)$ is the amount of time that has been spent to send the current feature of source m. Define $\Omega_m(\lambda)$ as the set of all states (δ, d) such that if $\Delta_m(t) = \delta$ and $d_m(t) = d$, then the optimal solution for (4.16) is to take the passive action at time t.

Definition 4.1 (Indexability) *[73] Bandit m is said to be indexable if, as the cost λ increases from $-\infty$ to ∞, the set $\Omega_m(\lambda)$ increases monotonically, i.e., $\lambda_1 \leq \lambda_2$ implies $\Omega_m(\lambda_1) \subseteq \Omega_m(\lambda_2)$. The RMAB problem* (4.5)-(4.8) *is said to be indexable if all $(M+1)$ bandits are indexable.*

Theorem 4.4 *The RMAB problem* (4.5)-(4.8) *is indexable.*

Proof 4.3 *See Appendix 4.C.*

Definition 4.2 (Whittle index) *[73] Given indexability, the Whittle index* $W_m(\delta, d)$ *of bandit* m *at state* (δ, d) *is*

$$W_m(\delta, d) = \inf\{\lambda \in \mathbb{R} : (\delta, d) \in \Omega_m(\lambda)\}. \tag{4.25}$$

Lemma 4.1 *The Whittle index of the dummy bandits is 0, i.e.,* $W_0(\delta, 0) = 0$ *for any* δ.

Theorem 4.5 *The following assertions are true for the Whittle index* $W_m(\delta, d)$ *of bandit* m *for* $m = 1, 2, \ldots, M$*:*

(a) If $d = 0$*, then for* $m = 1, \ldots, M$*,*

$$W_m(\delta, 0) = \max_{b \in \mathbb{Z}: 0 \le b \le B_m - 1} W_{m,b}(\delta, 0), \tag{4.26}$$

where

$$\begin{aligned} W_{m,b}(\delta, 0) =& \frac{1}{\mathbb{E}[T_{l,1}]}\, \mathbb{E}\Big[D_{m,i+1}(\gamma_m(\delta)) - D_{m,i}(\gamma_m(\delta))\Big]\, \gamma_m(\delta) \\ & - \frac{1}{\mathbb{E}[T_{l,1}]} \mathbb{E}\left[\sum_{t=D_{m,i}(\gamma_m(\delta))}^{D_{m,i+1}(\gamma_m(\delta))-1} w_m p_m(\Delta_m(t))\right], \end{aligned} \tag{4.27}$$

$\Delta_m(t) = t - S_{m,i}(\gamma_m(\delta)) + b$, $\gamma_m(\delta)$ *is defined in* (4.19), $D_{m,i+1}(\gamma_m(\delta)) = S_{m,i+1}(\gamma_m(\delta)) + T_{m,i}$, *and* $S_{m,i+1}(\gamma_m(\delta))$ *is given by*

$$S_{m,i+1}(\gamma_m(\delta)) = \min_{t \in \mathbb{Z}} \big\{t \ge D_{m,i}(\gamma_m(\delta)) : \gamma_m(\Delta_m(t)) \ge \gamma_m(\delta)\big\}. \tag{4.28}$$

(b) If $d > 0$*, then* $W_m(\delta, d) = -\infty$ *for* $m = 1, \ldots, M$.

Proof 4.4 *See Appendix 4.D.*

Theorem 4.5 presents an analytical expression of the Whittle index of bandit m for $m = 1, 2, \ldots, M$. If no feature of source m is being served by a channel at time slot t such

Algorithm 1 Scheduling Policy for Multi-source, Multi-channel Inference Error Minimization (4.2)-(4.4)

Initialize $t = 0$
Input the optimal dual variable λ^* of the problem (4.14).
for $t = 0, 1, \ldots$ **do**
    Update $\Delta_m(t)$ and $d_m(t)$ for all m.
    $W_m \leftarrow W_m(\Delta_m(t), d_m(t))$ for all m.
    for all channel $n = 1, 2, \ldots, N$ **do**
        if channel n is idle and $\max_l W_l > 0$ **then**
            Schedule source $m = \arg\max_l W_l$.
            Send the feature from the position $b_m^*(\lambda^*)$ in
            source m's buffer.
            $W_m \leftarrow -\infty$.
        end if
    end for
    $t \leftarrow t + 1$.
end for

that $d_m(t) = 0$, then the Whittle index of bandit m at time slot t is determined by (4.26). Otherwise, if source m is being served by a channel at time slot t such that $d_m(t) > 0$, then the Whittle index of bandit m at time t is $-\infty$.

In the special case that (i) the AoI function $p_m(\cdot)$ is non-decreasing and (ii) $T_{m,i} = 1$, it holds that $\gamma_m(\delta) = w_m p_m(\delta + 1)$ and for $\delta = 1, 2, \ldots,$

$$W_m(\delta, 0) = w_m \left[\delta \; p_m(\delta + 1) - \sum_{k=1}^{\delta} p_m(k) \right]. \tag{4.29}$$

By this, the Whittle index in Section IV of [26, Equation (7)] is recovered from Theorem 4.5.

Let $A(t)$ denote the number of available channel at the beginning of time slot t, where $A(t) \leq N$. Then, $A(t)$ bandits with the highest Whittle index are activated at any time slot t. As stated in Lemma 4.1, all N dummy bandits have Whittle index of $W_0(\Delta_0(t), d_0(t)) = 0$. Consequently, if a bandit m (for $m = 1, 2, ..., M$) possesses a negative Whittle index, denoted as $W_m(\Delta_m(t), d_m(t)) < 0$, it will remain inactive.

Now, we return to the original RMAB (4.2)-(4.4) with no dummy bandits. As illustrated in Algorithm 1, if channel n is idle, then source m with highest non-negative Whittle index is activated.

λ^*-based Selection-from-Buffer Policy

When the m-th bandit is activated, our policy in Algorithm 1 sends the feature from the buffer position $b_m^*(\lambda^*)$, determined by

$$b_m^*(\lambda^*) = \arg \min_{b \in \{0,1,\ldots,B_m-1\}} \bar{p}_{m,b,opt}(\lambda^*), \tag{4.30}$$

where λ^* is the optimal solution to (4.14) and $\bar{p}_{m,b,opt}(\lambda)$ is the optimal objective value of (4.17).

Asymptotic Optimality of the Proposed Scheduling Policy

Let π_{our} denote the scheduling policy outlined in Algorithm 1. Now, we demonstrate that π_{our} is asymptotically optimal.

Definition 4.3 (Asymptotically optimal) *[73, 74] Initially, we have N channels and M sources. Let $\bar{p}_\pi^r$ represent the expected long-term average inference error under policy π, where both the number of channels rN and the number of bandits rM are scaled by r. The policy π_{our} will be asymptotically optimal if $\bar{p}_{\pi_{\text{our}}}^r \leq \bar{p}_\pi^r$ for all $\pi \in \Pi$ as r approaches ∞, while maintaining a constant ratio $\alpha = \frac{rM}{rN}$.*

In (4.9)-(4.12), we have $M+N$ bandits with M sources and N dummy bandits, and N channels. Two bandits are considered to be in the same class if they share identical penalty functions, weights, and transition probabilities. The dummy bandits belong to the same class. However, among the remaining M sources, no two sources share the same combination of penalty function, weight, and transition probabilities. Therefore, we start with $M+1$ distinct classes of bandits. Then, we increase the number of sources rM, number of dummy bandits rN, and number of channels rN with scaling factor r, while maintaining the ratio $\frac{rM}{rN}$.

The state of a class m bandit is represented by two tuple $(\Delta_m(t), d_m(t))$, where $\Delta_m(t)$ represents the AoI and $d_m(t)$ is the amount of time spent to send the current feature from a class m bandit. Let $\mu_m(t) \in \{0, 1, \ldots, B_m\}$ be the action taken for a class m bandit. If $\mu_m(t) = 0$, no feature is selected for transmission; otherwise, if $\mu_m(t) = b$, a feature from buffer position $(b-1) \in \{0, 1, \ldots, B_m - 1\}$ is selected for transmission. Given state (δ, d) and action μ of a class m bandit, we denote by $P^{m,\mu}_{(\delta',d'),(\delta,d)}$ the transition probability to a state (δ', d').

Because we assume that the transmission times $T_{m,i}$ are bounded for all m and i, we can find a $d_{\text{bound}} \in \mathbb{N}$ such that $0 < T_{m,i} \leq d_{\text{bound}}$ for all m and i. Then, the amount of time $d_m(t)$ that has been spent to send the current feature of source m by time slot t is also bounded and $d_m(t) \in \{0, 1, \ldots, d_{\text{bound}}\}$. Also, we observe from Figs. 2.1-2.5 that the inference error function $p_m(\delta)$ converges after a large AoI value. We can find an AoI δ_{bound} such that $p_m(\delta) = p_m(\delta_{\text{bound}})$ for all $\delta \geq \delta_{\text{bound}}$ and for all m.

We denote by $V^m_{\delta,d}(t)$ the fraction of class m bandits with $\Delta_m(t) = \delta$ and $d_m(t) = d$. Let $U^m_{\delta,d,\mu}(t)$ be the fraction of class m bandits with $\Delta_m(t) = \delta$, $d_m(t) = d$, and $\mu_m(t) = \mu$. We denote by $\tilde{V}^m_d(t)$ the fraction of class m bandits with $d_m(t) = d$ and $\Delta_m(t) > \delta_{\text{bound}}$. Moreover, let $\tilde{U}^m_{d,\mu}(t)$ be the fraction of class m bandits with $d_m(t) = d$, $\Delta_m(t) > \delta_{\text{bound}}$, and $\mu(t) = \mu$. We define variables $v^m_{\delta,d}$, $u^m_{\delta,d,\mu}$, $\tilde{v}^m_d$, and $\tilde{u}^m_{d,\mu}$ for all δ, d, μ, and m as follows:

$$v^m_{\delta,d} := \limsup_{T\to\infty} \sum_{t=0}^{T-1} \frac{1}{T} \mathbb{E}[V^m_{\delta,d}(t)], \tag{4.31}$$

$$u^m_{\delta,d,\mu} := \limsup_{T\to\infty} \sum_{t=0}^{T-1} \frac{1}{T} \mathbb{E}[U^m_{\delta,d,\mu}(t)], \tag{4.32}$$

$$\tilde{v}^m_d := \limsup_{T\to\infty} \sum_{t=0}^{T-1} \frac{1}{T} \mathbb{E}[\tilde{V}^m_d(t)], \tag{4.33}$$

$$\tilde{u}^m_{d,\mu} := \limsup_{T\to\infty} \sum_{t=0}^{T-1} \frac{1}{T} \mathbb{E}[\tilde{U}^m_{d,\mu}(t)]. \tag{4.34}$$

A channel is occupied by a bandit, if either $d > 0$ or $\mu > 0$. Then, the time-averaged expected fraction of class m-bandits occupied a channel is given by

$$c_m = \sum_{\substack{\delta<\delta_{\text{bound}},d>0 \\ \mu=0}} u^m_{\delta,d,\mu} + \sum_{\substack{\delta<\delta_{\text{bound}},d=0 \\ \mu>0}} u^m_{\delta,d,\mu} + \sum_{d>0,\mu=0} \tilde{u}^m_{d,\mu} + \sum_{d=0,\mu>0} \tilde{u}^m_{d,\mu}. \tag{4.35}$$

By using c_m, the time-average channel constraint (4.12) can be expressed as

$$\sum_{m=0}^{M} c_m = N. \tag{4.36}$$

We define the vectors $\mathbf{V}^m(t)$, $\mathbf{U}^m(t)$, $\mathbf{v}^m$, and $\mathbf{u}^m$ to contain $(V^m_{\delta,d}(t), \tilde{V}^m_d(t))$, $(U^m_{\delta,d,\mu}(t), \tilde{U}^m_{d,\mu}(t))$, $(v^m_{\delta,d}, \tilde{v}^m_d)$, and $(u^m_{\delta,d,\mu}, \tilde{u}^m_{d,\mu})$, respectively, for all $\delta = 1, 2, \ldots, \delta_{\text{bound}}$, $d = 0, 1, \ldots, d_{\text{bound}}$, and $\mu = 0, 1, \ldots, B_m$.

Now, we provide a uniform global attractor condition. For a policy π, we can have a mapping Ψ_π of the state transitions, given by

$$\Psi_\pi((\mathbf{v}^m)^M_{m=1}) := \mathbb{E}_\pi[(\mathbf{V}^m(t))^M_{m=1}(t+1) | (\mathbf{V}^m(t))^M_{m=1}(t) = (\mathbf{v}^m)^M_{m=1}]. \tag{4.37}$$

We define the t-th iteration of the maps $\Psi_{\pi,t\geq 0}(\cdot)$ as $\Psi_{\pi,0}((\mathbf{v}^m)^M_{m=1}) = (\mathbf{v}^m)^M_{m=1}$ and $\Psi_{\pi,t+1}((\mathbf{v}^m)^M_{m=1}) = \Psi_\pi(\Psi_{\pi,t}((\mathbf{v}^m)^M_{m=1}))$.

Definition 4.4 *[74]* ***Uniform Global attractor.*** *An equilibrium point $(\mathbf{v}^{m*})^M_{m=1}$ associated with the optimal solution of (4.9)-(4.12) is a uniform global attractor of $\Psi_{\pi,t\geq 0}(\cdot)$, i.e., for all $\epsilon > 0$, there exists $T(\epsilon) > 0$ such that for all $t \geq T(\epsilon)$ and for all $(\mathbf{v}^m)^M_{m=1}$, one has $\|\Psi_{\pi,t}((\mathbf{v}^m)^M_{m=1}) - (\mathbf{v}^{m*})^M_{m=1})\|_1 \leq \epsilon$.*

Theorem 4.6 *Under the uniform global attractor condition provided in Definition 4.4, π_{our} is asymptotically optimal.*

Proof 4.5 (Proof sketch) *We first establish that if m-th source is selected, then there exists an optimal feature selection policy that always selects features from the buffer position* $b_m^*(\lambda^*)$*. Hence, the multiple action RMAB problem* (4.2)-(4.4) *reduces to a binary action RMAB problem. Then, we use [74, Theorem 13] to prove Theorem 4.6. See Appendix 4.E for details.*

4.4 Data Driven Evaluations

In this section, we illustrate the performance of our scheduling policies, where the inference error function $p(\delta)$ is collected from the data driven experiments in Chapter 2.3. Now, we evaluate the following three multiple-source scheduling policies:

1. Maximum age first (MAF), Generate-at-will: At time slot t, if a channel is free, this policy schedules the freshest generated feature from source $\arg\max_{l\in A(t)} \Delta_l(t)$, where $A(t)$ is the set of available sources in time slot t.

2. Whittle index, Generate-at-will: Denote

$$l_0^* = \underset{l\in A(t)}{\text{argmax}} W_{l,0}(\Delta_l(t)). \tag{4.38}$$

   If a channel is free and $\max_{l\in A(t)} W_{l,0}(\Delta_l(t)) \geq 0$, the freshest feature from the source l_0^* is scheduled; otherwise, no source is scheduled.

3. Proposed Policy: The policy is described in Algorithm 1.

4. Lower bound: Given the optimal dual variable $\lambda = \lambda^*$, the lower bound is obtained by implementing policy $(f_m^*(\lambda^*), g_m^*(\lambda^*))_{m=0}^M$, which is defined in Theorem 4.3.

5. Upper bound: The upper bound is obtained if none of the sources is scheduled at every time slot t.

In Figs. 4.2-4.3, we present a comparative analysis of our proposed policy alongside other policies. The plots depict the performance evaluation in a scenario where the number of sources is set to $m = 500$. The inference error for half of the sources is illustrated in Fig.

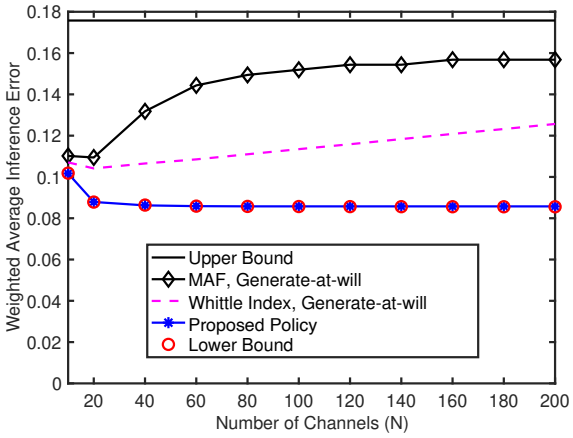


Figure 4.2: Time-average weighted inference error vs. number of channels (N).

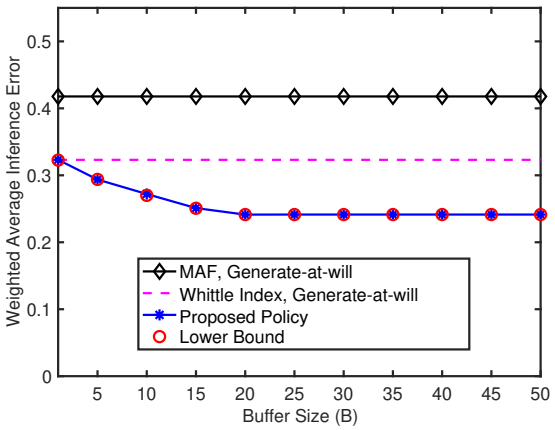


Figure 4.3: Time-average weighted inference error vs. buffer size (B).

2.3(c) and this corresponds to a weight of $w_1 = 1$; the inference error for the remaining half is depicted in Fig. 2.2(c), with a weight of $w_2 = 5$. In this section, the transmission time for all features from both classes of sources is set to 1 time slot.

In Fig. 4.2, we plot the weighted time-average inference error versus the number of channels, where the buffer size of all sources is set to 40 (i.e., $B_l = B = 40$ for all l). From Fig. 4.2, it is evident that our proposed policy outperforms the "Whittle index, Generate-at-will" and "MAF, Generate-at-will" policies. Specifically, our policy achieves a weighted average inference error that is twice as low as that of the "MAF, Generate-at-will" policy. Furthermore, as shown in Fig. 4.2, the performance of our policy matches

the lower bound of the multi-source, multi-action scheduling problem, thereby validating its asymptotic optimality.

Fig. 4.3 illustrates the weighted time-average inference error versus the buffer size B, with the number of channels set to $N = 50$. The results presented in Fig. 4.3 underscore the effectiveness of the "selection-from-buffer" model. The weighted time-average inference error achieved by our policy decreases as the buffer size B increases, eventually reaching a plateau at a buffer size of 20.

Appendix

4.A Proof of Theorem 4.3

In this section, we prove Theorem 4.2, and Theorem 4.3. These theorems provide optimal solutions for the scheduling problems (4.16) and (4.17). We begin by deriving the optimal solution for (4.16). Subsequently, the optimal solutions for (4.17) follows directly, as this problem is a special cases of (4.16).

Since the problem (4.16) focuses solely on a single source, we simplify the notation by omitting the source index m and rewrite the problem (4.16) as follows:

$$\bar{p}_{opt}(\lambda) = \inf_{\pi \in \Pi} \limsup_{T \to \infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} w\ p(\Delta(t)) + \lambda c(t)\right], \tag{4.39}$$

where $p(\Delta(t))$ is the penalty at time t, $\Delta(t) \in \mathbb{Z}^+$ is the AoI, $c(t) \in \{0, 1\}$ is the channel occupation status at time t, $\pi = ((S_1, b_1), (S_2, b_2), \ldots)$ is a scheduling policy, Π is the set of all causal and signal-agnostic scheduling policies, $w > 0$ is a weight, and $\bar{p}_{opt}(\lambda)$ is the optimal objective value to (4.39).

The scheduling problem (4.39) is an infinite-horizon average-cost semi-Markov decision process (SMDP) [69, Chapter 5.6]. We provide a detailed description of the components of this problem:

- **Decision Time:** Each i-th feature delivery time $D_i = S_i + T_i$ is a decision time of the problem (4.39), where S_i is the scheduling time of the i-th feature and the i-th feature takes $T_i \geq 1$ time slots to be delivered.

- **State:** At time slot D_i, the state of the system is represented by AoI $\Delta(D_i)$.

- **Action:** Let $\tau_{i+1} = S_{i+1} - D_i$ represent the waiting time for sending the $(i+1)$-th feature after the i-th feature is delivered. As we consider $S_0 = 0$ and $S_i = \sum_{j=1}^{i}(T_{j-1} + \tau_j)$ for each $i = 1, 2, \ldots$. Given $(T_0, T_1, \ldots)$, the sequence $(S_1, S_2, \ldots)$ is uniquely determined by $(\tau_1, \tau_2, \ldots)$. Hence, one can also use $(\tau_1, \tau_2, \ldots)$ to represent a sequence of actions instead of $(S_1, S_2, \ldots)$.

  At time D_i, the scheduler decides the waiting time τ_{i+1} and the buffer position b_{i+1}.

- **State Transitions:** The AoI process $\Delta(t)$ evolves as

$$\Delta(t) = \begin{cases} T_i + b_i, & \text{if } t = D_i, i = 0, 1, \ldots, \\ \Delta(t-1) + 1, & \text{otherwise.} \end{cases} \tag{4.40}$$

- **Expected Transition Time:** The expected time difference between two decision times, D_i and D_{i+1}, is given by

$$\begin{aligned} \mathbb{E}[D_{i+1} - D_i] &= \mathbb{E}[S_{i+1} + T_{i+1} - (S_i + T_i)] \\ &= \mathbb{E}[S_i + T_i + \tau_{i+1} + T_{i+1} - S_i - T_i] \\ &= \mathbb{E}[\tau_{i+1} + T_{i+1}]. \end{aligned} \tag{4.41}$$

- **Expected Transition Cost:** The expected cumulative cost incurred during the interval between two decision times, D_i and D_{i+1}, can be calculated as

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=D_i}^{D_{i+1}-1} \Big(w\ p(\Delta(t)) + \lambda c(t) \Big)\right] \\ =&\mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} w\ p(\Delta(D_i + k))\right] + \lambda \mathbb{E}[T_{i+1}]. \end{aligned} \tag{4.42}$$

The infinite-horizon average-cost SMDP (4.39) can be solved by using dynamic programming [69, 72]. Let $h : \mathbb{Z}^+ \mapsto \mathbb{R}$ be the relative value function associated with the average-cost SMDP (4.39). At time $t = D_i$, the optimal action (τ_{i+1}, b_{i+1}) can be determined by solving

the following Bellman optimality equation [69, P. 275]:

$$\begin{aligned} h(\Delta(D_i)) &= \inf_{\substack{\tau_{i+1}\in\{0,1,\dots\} \\ b_{i+1}\in\{0,\dots,B-1\}}} \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} wp(\Delta(D_i+k))\right] + \lambda\mathbb{E}[T_{i+1}] \\ &\quad - \bar{p}_{opt}(\lambda)\mathbb{E}[\tau_{i+1}+T_{i+1}] + \mathbb{E}[h(\Delta(D_{i+1})] \\ &= \inf_{\substack{\tau_{i+1}\in\{0,1,\dots\} \\ b_{i+1}\in\{0,\dots,B-1\}}} \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} \Big(wp(\Delta(D_i+k)) - \bar{p}_{opt}(\lambda)\Big)\right] \\ &\quad + \lambda\mathbb{E}[T_{i+1}] + \mathbb{E}[h(\Delta(D_{i+1})] \\ &= \inf_{\substack{\tau_{i+1}\in\{0,1,\dots\} \\ b_{i+1}\in\{0,\dots,B-1\}}} \mathbb{E}\left[\sum_{k=0}^{\tau_{i+1}+T_{i+1}-1} \Big(wp(\Delta(D_i+k)) - \bar{p}_{opt}(\lambda)\Big)\right] \\ &\quad + \lambda\mathbb{E}[T_{i+1}] + \mathbb{E}[h(T_{i+1}+b_{i+1})], \end{aligned} \tag{4.43}$$

where the last equality holds because $\Delta(D_{i+1}) = T_{i+1} + b_{i+1}$.

From (4.43), it is observed that the buffer position b_{i+1} only depends on the term $\mathbb{E}[h(T_{i+1}+b_{i+1})]$, while the waiting time τ_{i+1} has no impact on $\mathbb{E}[h(T_{i+1}+b_{i+1})]$. Hence, the optimal buffer position b^*_{i+1} is determined by

$$b^*_{i+1} = \arg\min_{b_{i+1}\in\{0,1,\dots,B-1\}} \mathbb{E}[h(T_{i+1}+b_{i+1})]. \tag{4.44}$$

Since T_i's are i.i.d., $\mathbb{E}[h(T_{i+1}+b)] = \mathbb{E}[h(T_i+b)] = \cdots = \mathbb{E}[h(T_1+b)]$ for all i and b. Hence, from (4.44), it is evident that there exists a $b^* \in \{0,1,\dots,B-1\}$ such that $b^*_1 = b^*_2 = \cdots = b^*_{i+1} = b^*$ that satisfies

$$b^* = \arg\min_{b\in\{0,1,\dots,B-1\}} \mathbb{E}[h(T_1+b)]. \tag{4.45}$$

Because the optimal buffer position is time-invariant, the problem (4.39) can be expressed as

$$\bar{p}_{opt}(\lambda) = \min_{b\in\{0,1,\dots,B-1\}} \bar{p}_{b,opt}(\lambda), \tag{4.46}$$

where $\bar{p}_{b,opt}(\lambda)$ is given by

$$\bar{p}_{b,opt}(\lambda) = \inf_{\pi_b \in \Pi_b} \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}_{\pi_b} \left[\sum_{t=0}^{T-1} w\ p(\Delta(t)) + \lambda c(t) \right], \tag{4.47}$$

$\pi_b = ((S_1, b), (S_2, b), \ldots)$, Π_b is the set of all causal and signal-agnostic scheduling policies π_b with fixed buffer position b, and $\bar{p}_{b,opt}(\lambda)$ is the optimal objective value to (4.47).

At every i-th decision time D_i of the average-cost SMDP (4.47), the scheduler decides the waiting time τ_{i+1}. The average-cost SMDP (4.47) can be solved by using dynamic programming [69,72]. Given AoI value δ at decision time D_i, the Bellman optimality equation of (4.47) is obtained by substituting $\Delta(D_i) = \delta$, $b_{i+1} = b$, and $\bar{p}_{opt}(\lambda) = \bar{p}_{b,opt}(\lambda)$ into (4.43), given by

$$\begin{aligned} h_b(\delta) = & \inf_{\tau\in\{0,1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_{i+1}-1} (w\ p(\delta+k) - \bar{p}_{b,opt}(\lambda)) \right] \\ & + \lambda\mathbb{E}[T_{i+1}] + \mathbb{E}[h_b(T_{i+1}+b)],\ \delta = 1, 2, \ldots \\ = & \inf_{\tau\in\{0,1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_{1}-1} (w\ p(\delta+k) - \bar{p}_{b,opt}(\lambda)) \right] \\ & + \lambda\mathbb{E}[T_{1}] + \mathbb{E}[h_b(T_{1}+b)],\ \delta = 1, 2, \ldots, \end{aligned} \tag{4.48}$$

where the last equality holds because T_i's are identically distributed. Let $\tau(\delta, \bar{p}_{b,opt}(\lambda))$ be an optimal solution to (4.48). If $\Delta(D_i) = \delta$, then an optimal waiting time τ_{i+1} of (4.47) for sending the $(i+1)$-th feature is $\tau(\delta, \bar{p}_{b,opt}(\lambda))$.

From (4.48), we can show that $\tau(\delta, \bar{p}_{b,opt}(\lambda)) = 0$ if

$$\begin{aligned} & \inf_{\tau\in\{1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_{1}-1} (w\ p(\delta+k) - \bar{p}_{b,opt}(\lambda)) \right] \\ & \geq \mathbb{E}\left[\sum_{k=0}^{T_1-1} (w\ p(\delta+k) - \bar{p}_{b,opt}(\lambda)) \right]. \end{aligned} \tag{4.49}$$

After some rearrangement, the inequality (4.49) can also be expressed as

$$\inf_{\tau\in\{1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau-1}(w\ p(\delta+k+T_1)-\bar{p}_{b,opt}(\lambda))\right]\geq 0. \tag{4.50}$$

Next, similar to [25, Lemma 7], the following lemma holds.

Lemma 4.2 *The inequality* (4.50) *holds if and only if*

$$\inf_{\tau\in\{1,2,\ldots\}} \frac{1}{\tau}\mathbb{E}\left[\sum_{k=0}^{\tau-1} w\ p(\delta+k+T_1)\right]\geq \bar{p}_{b,opt}(\lambda). \tag{4.51}$$

According to (4.19), the left-hand side of (4.51) is equal to $\gamma(\delta)$. Similarly, $\tau(\delta,\bar{p}_{b,opt}(\lambda))=1$, if $\tau(\delta,\bar{p}_{b,opt}(\lambda))\neq 0$ and

$$\inf_{\tau\in\{2,3,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau-1}(w\ p(\delta+k+T_1)-\bar{p}_{b,opt}(\lambda))\right]\geq 0 \tag{4.52}$$

By using Lemma 4.2 and (4.19), we can show that the inequality (4.52) can be rewritten as

$$\gamma(\delta+1)\geq \bar{p}_{b,opt}(\lambda). \tag{4.53}$$

By repeating this process, we get $\tau(\delta,\bar{p}_{b,opt}(\lambda))=k$ is optimal, if $\tau(\delta,\bar{p}_{b,opt}(\lambda))\neq 0,1,\ldots,k-1$ and

$$\gamma(\delta+k)\geq \bar{p}_{b,opt}(\lambda). \tag{4.54}$$

Hence, the optimal waiting time $\tau_{i+1}=\tau(\delta,\bar{p}_b(\lambda))$ is determined by

$$\tau(\delta,\bar{p}_b(\lambda))=\min_{k\in\mathbb{Z}}\{k\geq 0:\gamma(\delta+k)\geq \bar{p}_{b,opt}(\lambda)\}. \tag{4.55}$$

Now, we are ready to compute the optimal objective value $\bar{p}_{b,opt}(\lambda)$. Using (4.48), we can determine the value of $\mathbb{E}[h_b(T_1+b)]$, which is given by

$$\begin{aligned}\mathbb{E}[h_b(T_1+b)] =& \mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\bar{p}_{b,opt}(\lambda))+T_1-1}(wp(T_1+b+k)-\bar{p}_{b,opt}(\lambda))\right] \\ &+\lambda\mathbb{E}[T_1]+\mathbb{E}[h_b(T_1+b)], \end{aligned} \tag{4.56}$$

which yields

$$\mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\bar{p}_{b,opt}(\lambda))+T_1-1}\Big(wp(T_1+b+k)-\bar{p}_{b,opt}(\lambda)\Big)\right]+\lambda\mathbb{E}[T_1]=0. \tag{4.57}$$

Rearranging (4.57), we get

$$\begin{aligned}\bar{p}_{b,opt}(\lambda) &= \frac{\mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\bar{p}_{b,opt}(\lambda))+T_1-1} wp(T_1+b+k)\right]+\lambda\mathbb{E}[T_1]}{\mathbb{E}[\tau(T_1+b,\bar{p}_{b,opt}(\lambda))+T_1]} \\ &= \frac{\mathbb{E}\left[\sum_{t=D_i(\bar{p}_{b,opt}(\lambda))}^{D_{i+1}(\bar{p}_{b,opt}(\lambda))-1} wp\big(\Delta(t)\big)\right]+\lambda E[T_1]}{\mathbb{E}\left[D_{i+1}(\bar{p}_{b,opt}(\lambda))-D_i(\bar{p}_{b,opt}(\lambda))\right]}, \end{aligned} \tag{4.58}$$

where $D_{i+1}(\bar{p}_{b,opt}(\lambda)) = S_{i+1}(\bar{p}_{b,opt}(\lambda)) + T_{i+1}$ and

$$S_{i+1}(\bar{p}_{b,opt}(\lambda)) = \min_{t\geq 0}\{t \geq D_i(\bar{p}_{b,opt}(\lambda)) : \gamma(\Delta(t)) \geq \bar{p}_{b,opt}(\lambda)\}, \tag{4.59}$$

and the last equality holds due to (4.41) and (4.42).

Now, by combining (4.46), (4.55), and (4.58) and by substituting appropriate values of λ and w, we obtain optimal solution to (4.16) and (4.17).

Finally, we need to prove that

$$\mathbb{E}\left[\sum_{t=D_i(\beta)}^{D_{i+1}(\beta)-1} wp\big(\Delta(t)\big)\right]-\beta\mathbb{E}\left[D_{i+1}(\beta)-D_i(\beta)\right]+\lambda\mathbb{E}[T_1]=0. \tag{4.60}$$

has a unique root. We define

$$J(\beta) = \mathbb{E}\left[\sum_{t=D_i(\beta)}^{D_{i+1}(\beta)-1} wp\big(\Delta(t)\big)\right] - \beta\mathbb{E}\left[D_{i+1}(\beta) - D_i(\beta)\right] + \lambda\mathbb{E}[T_1]. \tag{4.61}$$

Lemma 4.3 *The function $J(\beta)$ has the following properties:*

(i) The function $J(\beta)$ is concave, continuous, and strictly decreasing in β.

(ii) $\lim_{\beta\to\infty} j(\beta) = -\infty$ and $\lim_{\beta\to-\infty} j(\beta) = \infty$.

Proof 4.6 *See Appendix 4.B*

Lemma 4.3 proves the uniqueness of (4.63). Also, the uniqueness of the root of (4.20) follows immediately from Lemma 4.3.

4.B Proof of Lemma 4.3

We denote $D_{i+1}(\beta) = S_{i+1}(\beta) + T_{i+1}$, $S_{i+1}(\beta) = \tau(\Delta(D_i), \beta) + D_i(\beta)$, where $\tau(\delta, \beta)$ is defined as the optimal solution of

$$\inf_{\tau\in\{0,1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} (wp(\delta+k) - \beta)\right]. \tag{4.62}$$

Because T_i's are i.i.d., and $\Delta(D_i) = T_i + b$, we can express (4.61) as

$$\begin{aligned} J(\beta) &= \mathbb{E}\left[\sum_{t=D_i(\beta)}^{D_{i+1}(\beta)-1} wp\big(\Delta(t)\big)\right] - \beta\mathbb{E}\left[D_{i+1}(\beta) - D_i(\beta)\right] + \lambda\mathbb{E}[T_1] \\ &= \mathbb{E}\left[\sum_{k=0}^{\tau(T_1+b,\beta)+T_1-1} (wp(T_1+b+k) - \beta)\right] + \lambda\mathbb{E}[T_1] \\ &= \inf_{\tau\in\{0,1,\ldots} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} (wp(T_1+b+k) - \beta)\right] + \lambda\mathbb{E}[T_1]. \end{aligned} \tag{4.63}$$

Since the right-hand side of (4.63) is the pointwise infimum of the linear decreasing functions of β, $J(\beta)$ is concave, continuous, and strictly decreasing in β. This completes the proof of part (i) of Lemma 4.3.

Part (ii) of Lemma 4.3 holds because for any $\tau \geq 0$

$$\lim_{\beta \to \infty} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} (wp(T_1+b+k)-\beta)\right] = -\infty \tag{4.64}$$

and

$$\lim_{\beta \to -\infty} \mathbb{E}\left[\sum_{k=0}^{\tau+T_1-1} (wp(T_1+b+k)-\beta)\right] = \infty. \tag{4.65}$$

This completes the proof.

4.C Proof of Theorem 4.4

According to (4.22) and the definition of set $\Omega_m(\lambda)$, a point $(\Delta_m(t), d_m(t)) \in \Omega_m(\lambda)$ if either (i) $d_m(t) > 0$ such that a feature from source m is currently in service at time t, or (ii) $\gamma_m(\Delta_m(t)) < \bar{p}_{m,opt}(\lambda)$ such that the threshold condition in (4.22) for sending a new feature is not satisfied. By this, an analytical expression of set $\Omega_m(\lambda)$ is derived as

$$\Omega_m(\lambda) = \{(\delta, d) : d > 0 \text{ or } \gamma_m(\delta) < \bar{p}_{m,opt}(\lambda)\}, \tag{4.66}$$

where according to Theorem 4.2 and Theorem 4.3, $\beta = \bar{p}_{m,opt}(\lambda)$ is the unique root of

$$J_{m,1}(\beta) + \lambda \mathbb{E}[T_{m,1}] = 0, \tag{4.67}$$

and

$$J_{m,1}(\beta) = \mathbb{E}\left[\sum_{t=D_{m,i}(\beta)}^{D_{m,i+1}(\beta)-1} w_m p_m\big(\Delta_m(t)\big)\right] - \beta \mathbb{E}\left[D_{m,i+1}(\beta) - D_{m,i}(\beta)\right].$$

Because $\lambda \mathbb{E}[T_{m,1}]$ does not change with β, from Lemma 4.3, we can show that $J_{m,1}(\beta)$ is a strictly decreasing function of β with $\lim_{\beta \to \infty} j_{m,1}(\beta) = -\infty$ and $\lim_{\beta \to -\infty} j_{m,1}(\beta) = \infty$. Hence, the inverse function $J_{m,1}^{-1}(\cdot)$ exists and from (4.67), we get $J_{m,1}^{-1}(\lambda \mathbb{E}[T_{m,1}]) = \bar{p}_{m,opt}(\lambda)$. Because the inverse function of a strictly decreasing function is strictly increasing, $\bar{p}_{m,opt}(\lambda)$

is strictly increasing function of λ. Substituting this into (4.66), we get that if $\lambda_1 \leq \lambda_2$, then $\Omega_m(\lambda_1) \subseteq \Omega_m(\lambda_2)$.

For dummy bandits, it is optimal in (4.15) to activate a dummy bandit when $\lambda \leq 0$. Hence, dummy bandits are always indexable.

4.D Proof of Theorem 4.5

By substituting (4.66) into (4.25), we obtain, if $d > 0$, then $W_m(\delta, d) = -\infty$, and if $d = 0$, then

$$W_m(\delta, 0) = \inf\{\lambda : \gamma_m(\delta) < \bar{p}_{m,opt}(\lambda)\}. \tag{4.68}$$

Using (4.23) and (4.68), we get

$$W_m(\delta, 0) = \max_{0 \leq b \leq B_m - 1} W_{m,b}(\delta, 0), \tag{4.69}$$

where

$$W_{m,b}(\delta, 0) = \inf\{\lambda : \gamma_m(\delta) < \bar{p}_{m,b,opt}(\lambda)\}. \tag{4.70}$$

Because $\bar{p}_{m,b,opt}(\lambda)$ is strictly increasing function of λ, (4.70) implies that $W_{m,b}(\delta, 0)$ is unique and satisfies

$$\bar{p}_{m,b,opt}(W_{m,b}(\delta, 0)) = \gamma_m(\delta). \tag{4.71}$$

By including source index m into (3.32), we get

$$\bar{p}_{m,b,opt}(\lambda) = \frac{\mathbb{E}\left[\sum_{t=D_{m,i}(\bar{p}_{m,b,opt}(\lambda))}^{D_{m,i+1}(\bar{p}_{m,b,opt}(\lambda))-1} p_m\big(\Delta_m(t)\big)\right] + \lambda \mathbb{E}[T_{m,1}]}{\mathbb{E}\left[D_{i+1}(\bar{p}_{m,b,opt}(\lambda)) - D_i(\bar{p}_{m,b,opt}(\lambda))\right]}. \tag{4.72}$$

By substituting $\lambda = W_{m,b}(\delta, 0)$ and (4.71) into (4.72). and then, re-arranging, we get (4.27). This concludes the proof.

4.E Proof of Theorem 4.6

Preliminaries

Policy π in (4.9)-(4.12) can be expressed as a sequence of actions $\pi = (\mu(t))_{t=0,1,\ldots}$, with $\mu(t) = \mu_m(t))_{m=0}^{M}$ representing actions taken at successive time slots. Let $(\mu^*(t))_{t=0,1,\ldots}$ denote the optimal policy for solving (4.9)-(4.12). Given the optimal dual variable $\lambda = \lambda^*$, Theorems 4.2-4.3 imply that in the optimal policy for (4.9)-(4.12), class m bandits choose either action $\mu_m(t) = 0$ or $\mu_m(t) = b_m^*(\lambda^*) + 1$ at every time slot t. Therefore, the optimal state-action frequency satisfies: $u_{\delta,d,\mu}^{m*} = 0$ for all $\mu \neq 0, b_m^*(\lambda^*) + 1$. Hence, for all class m, only two actions can occur for every bandit in class m. Thus, our multiple-action RMAB problem can be reduced to a binary action RMAB problem. After that, for the truncated state space $(\delta, d) \in \{1, 2, \ldots, \delta_{\text{bound}}\} \times \{0, 1, \ldots, d_{\text{bound}}\}$, we directly use [74, Theorem 13 and Proposition 14] to prove the asymptotic optimality of our policy. Though [74, Theorem 13 and Proposition 14] is proved for a single class of bandits, we can use the results for multiple classes of bandits because of a similar argument provided in [77, Section 5]: we argue that having multiple classes of bandits can be represented by a single class of bandits by considering a larger state space: the state of a bandit would be (m, δ, d), where m is its class.

Chapter 5

Learning and Communications Co-design for Remote Inference: Feature Length Selection and Transmission Scheduling

5.1 Introduction

In this paper, we study a learning and communications co-design problem that jointly controls the timeliness and the length of the feature sequences. The contributions of this paper are summarized as follows:

- We demonstrated that the inference error is a function of the AoI, whereas the function is not necessarily monotonic. This chapter further investigates the impact of feature length on inference error. Our information-theoretic and experimental analysis show that the inference error is a non-increasing function of the feature length (See Figs. 2.5(a)-5.3(a), and Lemma 5.1).

- We propose a novel learning and communications co-design framework (see Section 5.2). In this framework, we adopted the "selection-from-buffer" model proposed in [12], which is more general than the popular "generate-at-will" model that was proposed in [20] and named in [21]. In addition, we consider both time-invariant and time-variant feature length. Earlier studies, for example [12, 78], did not consider time-variant feature length.

- For a single sensor-predictor pair and a single channel, this paper jointly optimizes feature length selection and transmission scheduling to minimize the time-averaged inference error. This joint optimization is formulated as an infinite time-horizon average-cost *semi-Markov decision process* (SMDP). Such problems often lack analytical solutions or closed-form expressions. Nevertheless, we are able to derive a closed-form expression for an optimal scheduling policy in the case of time-invariant feature length (Theorem 5.1). The optimal scheduling time strategy is a *threshold-based policy*. Our threshold-based scheduling approach differs significantly from previous threshold-based policies in e.g., [12, 23, 25–27], because our threshold function depends on both the AoI

value and the feature length, while prior threshold functions rely solely on the AoI value. In addition, our threshold function is *not necessarily monotonic* with AoI. This is a significant difference with prior studies [23, 25–27].

- We provide an optimal policy for the case of time-variant feature length. Specifically, Theorem 5.2 presents the Bellman equation for the average-cost SMDP with time-variant feature length. The Bellman equation can be solved by applying either relative value iteration or policy iteration algorithms [72, Sec. 11.4.4]. Given the complexity associated with converting the average-cost SMDP into a Markov Decision Process (MDP) suitable for relative value iteration, we opt for the alternative: using the *policy iteration* algorithm to solve our average-cost SMDP. By leveraging specific structural properties of the SMDP, we can simplify the policy iteration algorithm to reduce its computational complexity. The simplified policy iteration algorithm is outlined in Algorithm 2 and Algorithm 3.

- Furthermore, we investigate the learning and communications co-design problem for multiple sensor-predictor pairs and multiple channels. This problem is a restless multi-armed, multi-action bandit problem that is known to be PSPACE-hard [75]. Moreover, proving indexability condition relating to Whittle index policy [33] for our problem is fundamentally difficult. To this end, we propose a new scheduling policy named "Net Gain Maximization" that does not need to satisfy the indexability condition (Algorithm 5).

- Numerical evaluations demonstrate that our policies for the single source case can achieve up to 10000 times performance gain compared to periodic updating and zero-wait policy (see Figs. 5.5-5.6). Furthermore, our proposed multiple source policy outperforms the maximum age-first policy (see Fig. 5.7) and is close to a lower bound (see Fig. 5.8).

5.1.1 Related Works

The age of information (AoI) has emerged as a popular metric for analyzing and optimizing communication networks [32, 79], control systems [27, 80], remote estimation

[25, 45], and remote inference [11, 12]. As surveyed in [43], several studies have investigated sampling and scheduling policies for minimizing linear and nonlinear functions of AoI [12,21,23,26–28,30–32,79,81–84]. In most previous works [21,23,26–28,30–32,79,81–84], monotonic AoI penalty functions are considered. However, in a recent study [12], it is demonstrated that the monotonic assumption is not always true for remote inference. In contrast, the inference error is a function of AoI, but the function is not necessarily monotonic. The present paper further investigates the impact of feature length on the inference error and jointly optimizes AoI and feature length.

In recent years, researchers have increasingly employed information-theoretic metrics to evaluate information freshness [11, 12, 23, 24, 36–38, 85]. In [23, 24, 36], the authors utilized Shannon's mutual information to quantify the amount of information carried by received data messages about the current source value, and used Shannon's conditional entropy to measure the uncertainty about the current source value after receiving these messages. These metrics were demonstrated to be monotonic functions of the AoI when the source follows a time-homogeneous Markov chain [23, 24]. Built upon these findings, the authors of [38] extended this framework to include hidden Markov model. Furthermore, a Shannon's conditional entropy term $H_{\text{Shannon}}(Y_t|X_{t-\Delta(t)} = x)$ was used in [37, 85] to quantify information uncertainty. However, a gap still existed between these information-theoretic metrics and the performance of real-time applications such as remote estimation or remote inference. In our recent works [11,12,14] and the present paper, we have bridged this gap by using a generalized conditional entropy associated with a loss function L, called L-conditional entropy, to measure (or approximate) training and inference errors in remote inference, as well as the estimation error in remote estimation. For example, when the loss function $L(y, \hat{y})$ is chosen as a quadratic function $||y - \hat{y}||_2^2$, the L-conditional entropy $H_L(Y_t|X_{t-\Delta(t)}) = \min_\phi \mathbb{E}[(Y_t - \phi(X_{t-\Delta(t)}))^2]$ is exactly the minimum mean squared estimation error in remote estimation. This approach allows us to analyze how the AoI $\Delta(t)$ affects inference and estimation errors directly, instead of relying on information-theoretic metrics as intermediaries for assessing application performance. It is worth noting that Shannon's conditional entropy is a special case of L-conditional entropy, corresponding to the inference and estimation errors for softmax regression and maximum likelihood estimation, as discussed in Section 5.2.

The optimization of linear and non-linear functions of AoI for multiple source scheduling can be formulated as a restless multi-armed bandit problem [12, 26, 86–88]. Whittle, in his seminal work [33], proposed an index-based policy to address restless multi-armed bandit (RMAB) problems with binary actions determining whether to select or not select a bandit. Our multiple source scheduling problem is a RMAB problem with multiple actions. An extension of the Whittle index policy for multiple actions was provided in [89], but it requires to satisfy a complicated *indexability* condition. In [78], the authors considered joint feature length selection and transmission scheduling, where the penalty function was assumed to be non-decreasing in the AoI, the feature length is time-invariant, and there is only one communication channel. Under these assumptions, [78] established the indexability condition and developed a Whittle Index policy. Compared to [78], our work could handle both monotonic and non-monotonic AoI penalty functions, both time-invariant and time-variant feature lengths, and both one and multiple communication channels.

Because of (i) the time-variant feature length and non-monotonic AoI penalty function and (ii) the fact that there exist multiple transmission actions, we could not utilize the Whittle index theory to establish indexability for our multiple source scheduling problem. To address this challenge, we propose a new "Net Gain Maximization" algorithm (Algorithm 5) for multi-source feature length selection and transmission scheduling, which does not require indexability. During the revision of this paper, we found a related study [85], where the authors introduced a similar gain index-based policy for a RMAB problem with two actions: to send or not to send. The "Net Gain Maximization" algorithm that we propose is more general than the gain index-based policy in [85] due to its capacity to accommodate more than two actions in the RMAB.

5.2 System Model and Scheduling Policy

We consider a remote inference system composed of a sensor, a transmitter, and a receiver, as illustrated in Fig. 5.1. The sensor observes a time-varying target $Y_t \in \mathcal{Y}$ and feeds its measurement $V_t \in \mathcal{V}$ to the transmitter. The transmitter generates features from the sensory outputs and progressively transmits the features to the receiver through a

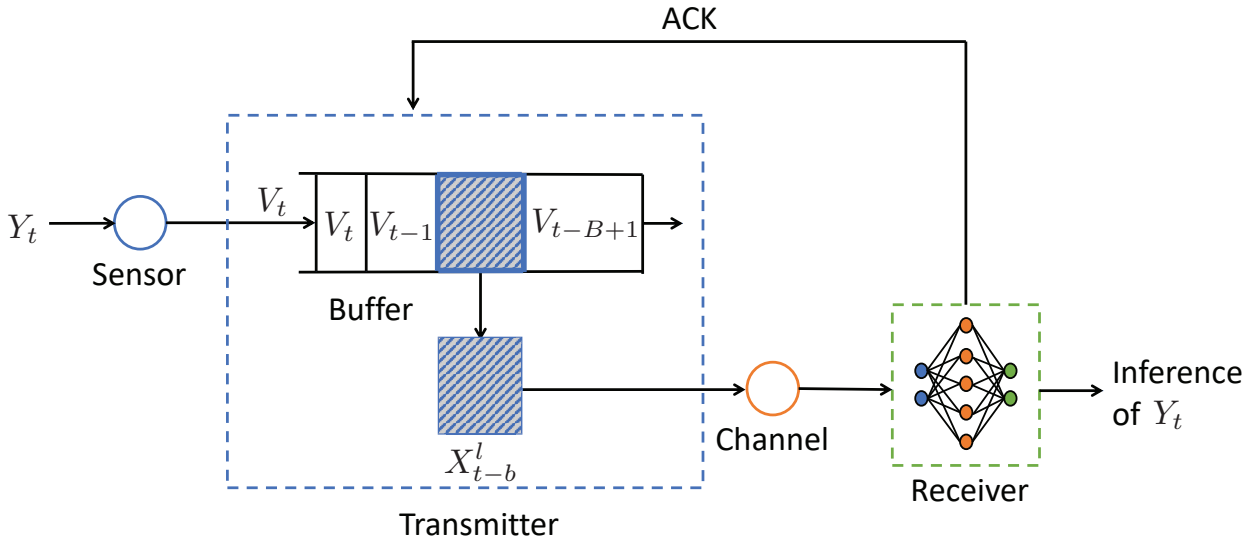


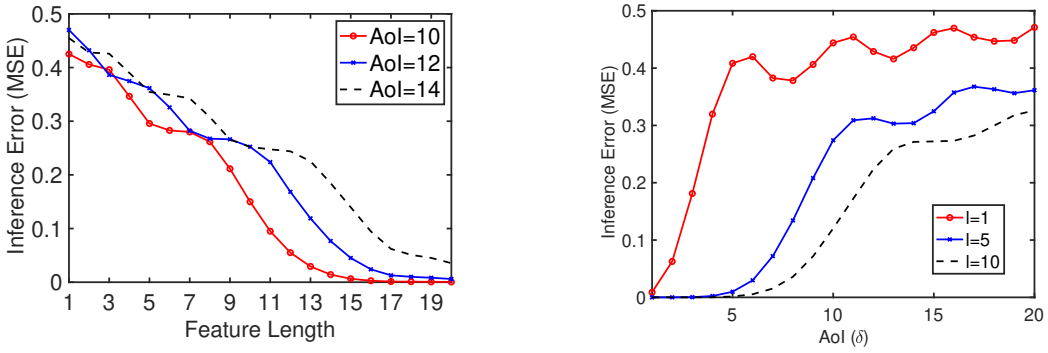
Figure 5.1: A remote inference system, where $X^l_{t-b} := (V_{t-b}, V_{t-b-1}, \ldots, V_{t-b-l+1})$ is a feature with sequence length l.

communication channel. Within the receiver, a neural network infers the time-varying target based on the received features.

System Model

The system is time-slotted and starts to operate at time slot $t = 0$. At every time slot t, the transmitter appends the sensory output $V_t \in \mathcal{V}$ to a buffer that stores the B most recent sensory outputs $(V_t, V_{t-1}, \ldots, V_{t-B+1})$; meanwhile, the oldest output V_{t-B} is removed from the buffer. We assume that the buffer is full initially, containing B signal values $(V_0, V_1, \ldots, V_{-B+1})$ at time $t = 0$. This ensures that the buffer remains consistently full at any time t.[1] The transmitter progressively generates a feature X^l_{t-b}, where each feature $X^l_{t-b} := (V_{t-b}, \ldots, V_{t-b-l+1}) \in \mathcal{V}^l$ is a temporal sequence of sensory outputs taken from the buffer such that $\mathcal{V}^l$ is the set of all l-tuples that take values from $\mathcal{V}$, $1 \leq l \leq B$, and $0 \leq b \leq B - l$. For ease of presentation, the temporal sequence length l of feature X^l_{t-b} is called *feature length* and the starting position b of feature X^l_{t-b} in the buffer is called *feature position*. If the channel is idle in time slot t, the transmitter can submit the feature X^l_{t-b} to the channel. Due to communication delays and channel errors, the feature is not instantly

[1]This assumption does not introduce any loss of generality. If the buffer is no full at time $t = 0$, it would not affect our results.



(a) Inference error vs. Feature length

(b) Inference error vs. AoI

Figure 5.2: Performance of wireless channel state information prediction: (a) Inference error Vs. Feature length and (b) Inference error Vs. AoI.

received. The most recently received feature is denoted as $X_{t-\delta}^{l} = (V_{t-\delta}, V_{t-\delta-1}, \ldots, V_{t-\delta-l+1})$, where the latest observation $V_{t-\delta}$ in feature $X_{t-\delta}^{l}$ is generated δ time slots ago. We call δ the *age of information (AoI)* which represents the difference between the time stamps of the target Y_t and the latest observation $V_{t-\delta}$ in feature $X_{t-\delta}^{l}$.

The receiver consists of B trained neural networks, each associated with a specific feature length $l = 1, 2, \ldots, B$. The neural network associated with feature length l takes the AoI $\delta \in \mathbb{Z}^+$ and the feature $X_{t-\delta}^{l} \in \mathcal{V}^l$ as inputs and generates an output $a = \phi_l(\delta, X_{t-\delta}^{l}) \in \mathcal{A}$, where the neural network is represented by the function $\phi_l : \mathbb{Z}^+ \times \mathcal{V}^l \mapsto \mathcal{A}$. The performance of the neural network is measured by a loss function $L : \mathcal{Y} \times \mathcal{A} \mapsto \mathbb{R}$, where $L(y, a)$ indicates the incurred loss if the output $a \in \mathcal{A}$ is used for inference when $Y_t = y$. The loss function L is determined by the purpose of the application. For example, in softmax regression (i.e., neural network based maximum likelihood classification), the output $a = Q_Y$ is a distribution of Y_t and the loss function $L_{\log}(y, Q_Y) = -\log Q_Y(y)$ is the negative log-likelihood of the value $Y_t = y$. In neural network based mean-squared estimation, a quadratic loss function $L_2(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ is used, where the action $a = \hat{\mathbf{y}}$ is an estimate of the target value $Y_t = \mathbf{y}$ and $\|\mathbf{y}\|_2$ is the euclidean norm of the vector $\mathbf{y}$.

Inference Error

We assume that $\{(Y_t, X_t^l), t \in \mathbb{Z}\}$ is a stationary process and the processes $\{(Y_t, X_t^l), t \in \mathbb{Z}\}$ and $\{\Delta(t), t = 0, 1, 2, \ldots\}$ are independent for every l. Under these assumptions, given

AoI δ and feature length l, the expected inference error is a function of δ and l, given by

$$\mathrm{err}_{\mathrm{inference}}(\delta, l) := \mathbb{E}_{Y, X^l \sim P_{Y_t, X_{t-\delta}^l}} \left[L\left(Y, \phi_l\left(\delta, X^l\right)\right)\right], \tag{5.1}$$

where $P_{Y_t, X_{t-\delta}^l}$ is the joint distribution of the label Y_t and feature $X_{t-\delta}^l$ during online inference and the function ϕ_l represents any trained neural network that maps from $\mathbb{Z}^+ \times \mathcal{V}^l$ to $\mathcal{A}$. The inference error $\mathrm{err}_{\mathrm{inference}}(\delta, l)$ can be evaluated through machine learning experiments.

In this paper, we conduct two experiments: (i) wireless channel state information (CSI) prediction and (ii) actuator states prediction in the OpenAI CartPole-v1 task [2]. Detailed information regarding the experimental setup for both experiments can be found in Appendix A of the supplementary material. The code for these experiments is available in GitHub repositories[2,3].

The experimental results, presented in Figs. 5.2(a)-5.3(a), demonstrate that the inference error decreases with respect to feature length. Moreover, Figs. 5.2(b)-5.3(b) illustrate that the inference error is not necessarily a monotonic function of AoI. These findings align with machine learning experiments conducted in [11, 12, 14]. Collectively, the results from this paper and those in [11,12,14] indicate that longer feature lengths can enhance inference accuracy and fresher features are not always better than stale features in remote inference.

Feature Length Selection and Transmission Scheduling Policy

Because (i) fresh feature is not always better than stale feature and (ii) longer feature can improve inference error, we adopted "selection-from-buffer" model, which is recently proposed in [12]. In contrast to the "generate-at-will" model [20,21], where the transmitter can only select the most recent sensory output V_t, the "selection-from-buffer" model offers greater flexibility by allowing the transmitter to pick multiple sensory outputs (which can be stale or fresh). In other words, "selection-from-buffer" model allows the transmitter to choose feature position b and feature length l under the constraints $1 \leq l \leq B-1$ and $0 \leq b \leq B-l$. Feature length selection represents a trade-off between learning and

[2] `https://github.com/Kamran0153/Channel-State-Information-Prediction`
[3] `https://github.com/Kamran0153/Impact-of-Data-Freshness-in-Learning`

communications: A longer feature can provide better learning performance (see Figs. 2.5-5.3), whereas it requires more channel resources (e.g., more time slots or more frequency resources) for sending the feature. This motivated us to study a learning-communication co-design problem that jointly optimizes the feature length, feature position, and transmission scheduling.

The feature length and feature position may vary across the features sent over time. Feature transmissions over the channel are non-preemptive: the channel must finish sending the current feature, before becoming available to transmit the next feature. Suppose that the i-th feature $X_{S_i-b_i}^{l_i} = (V_{S_i-b_i}, V_{S_i-b_i-1}, \ldots, V_{S_i-b_i-l_i+1})$ is submitted to the channel at time slot $t = S_i$, where l_i is its feature length and b_i is its feature position such that $1 \leq l_i \leq B$ and $0 \leq b_i \leq B - l_i$. It takes $T_i(l_i) \geq 1$ time slots to send the i-th feature over the channel. The i-th feature is delivered to the receiver at time slot $D_i = S_i + T_i(l_i)$, where $S_i < D_i \leq S_{i+1}$. The feature transmission time $T_i(l_i)$ depends on the feature length l_i. Due to time-varying channel conditions, we assume that, given feature length $l_i = l$, the $T_i(l)$'s are *i.i.d.* random variables, with a finite mean $1 \leq \mathbb{E}[T_i(l)] < \infty$. Once a feature is delivered, an acknowledgment (ACK) is sent back to the transmitter, notifying that the channel has become idle.

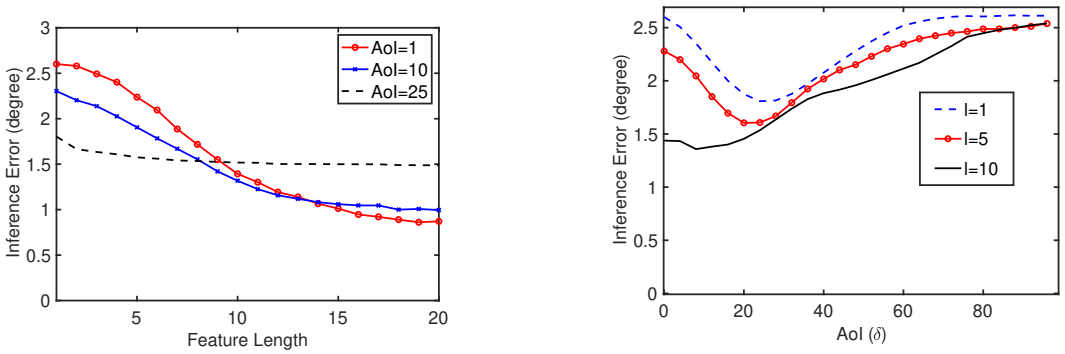
In time slot t, the $i(t)$-th feature $X_{S_{i(t)}-b_{i(t)}}^{l_{i(t)}}$ is the most recently received feature, where $i(t) = \max_i\{D_i \leq t\}$. The receiver feeds the feature $X_{S_{i(t)}-b_{i(t)}}^{l_{i(t)}}$ to the neural network to infer Y_t. We define *age of information (AoI)* $\Delta(t)$ is defined as the difference between the time-stamp of the freshest sensory output $V_{S_{i(t)}-b_{i(t)}}$ in feature $X_{S_{i(t)}-b_{i(t)}}^{l_{i(t)}}$ and the current time t, i.e.,

$$\Delta(t) := t - \max_i\{S_i - b_i : D_i \leq t\}. \tag{5.2}$$

Because $D_i < D_{i+1}$, it holds that

$$\Delta(t) = t - S_i + b_i, \text{ if } D_i \leq t < D_{i+1}. \tag{5.3}$$

The initial state of the system is assumed to be $S_0 = 0, l_0 = 1, b_0 = 0, D_0 = T_0(l_0)$, and $\Delta(0)$ is a finite constant.



(a) Inference error vs. Feature length

(b) Inference error vs. AoI

Figure 5.3: Performance of actuator state prediction in the OpenAI CartPole-v1 task under mechanical response delay: (a) Inference error Vs. Feature length and (b) Inference error Vs. AoI.

Let $\pi = ((S_1, b_1, l_1), (S_2, b_2, l_2), \ldots)$ represent a scheduling policy and Π denote the set of all the causal scheduling policies that satisfy the following conditions: (i) the scheduling time S_i, the feature position b_i, and the feature length l_i are decided based on the current and the historical information available at the transmitter such that $1 \leq l_i \leq B$ and $0 \leq b_i \leq B - l_i$, (ii) the scheduler does not have access to the realization of the process $\{(Y_t, X_t^l), t = 0, 1, 2, \ldots\}$, and (iii) the scheduler has access to the inference error function $\text{err}_{\text{inference}}(\cdot)$ and the distribution of $T_i(l)$ for each $l = 1, 2, \ldots, B$. We use $\Pi_{\text{inv}} \subset \Pi$ to denote the set of causal scheduling policies with time-invariant feature length, defined as

$$\Pi_{\text{inv}} := \bigcup_{l=1}^{B} \Pi_l, \tag{5.4}$$

where $\Pi_l := \{\pi \in \Pi : l_1 = l_2 = \cdots = l\}$.

5.3 Preliminaries: Impacts of Feature Length and AoI on Inference Error

In this section, we adopt an information-theoretic approach that was developed recently in [12] to show the impact of feature length l and AoI δ on the inference error $\text{err}_{\text{inference}}(\delta, l)$.

Information-theoretic Metrics for Training and Inference Errors

Training error $\text{err}_{\text{training}}(\delta, l)$ is expressed as a function of δ and l, given by

$$\text{err}_{\text{training}}(\delta, l) = \mathbb{E}_{Y, X^l \sim P_{\tilde{Y}_0, \tilde{X}^l_{-\delta}}}[L(Y, \phi_l(\delta, X^l))], \tag{5.5}$$

where ϕ_l a trained neural network used in (5.1) and $P_{\tilde{Y}_0, \tilde{X}^l_{-\delta}}$ is the joint distribution of the target $\tilde{Y}_0$ and the feature $\tilde{X}^l_{-\delta}$ in the training dataset. The training error $\text{err}_{\text{training}}(\delta, l)$ is lower bounded by

$$H_L(\tilde{Y}_0 | \tilde{X}^l_{-\delta}) = \min_{\phi_l \in \Phi} \mathbb{E}_{Y, X^l \sim P_{\tilde{Y}_0, \tilde{X}^l_{-\delta}}}[L(Y, \phi_l(\delta, X^l))], \tag{5.6}$$

where $\Phi = \{\phi_l : \mathbb{Z}^+ \times \mathcal{V}^l \mapsto \mathcal{A}\}$ is the set of all functions that map from $\mathbb{Z}^+ \times \mathcal{V}^l$ to $\mathcal{A}$. Because the trained neural network ϕ_l in (5.5) satisfies $\phi_l \in \Phi$, $H_L(\tilde{Y}_0 | \tilde{X}^l_{-\delta}) \leq \text{err}_{\text{training}}(\delta, l)$.

The inference error $\text{err}_{\text{inference}}(\delta, l)$ can be approximated as the following L-conditional cross entropy

$$H_L(P_{Y_t | X^l_{t-\delta}}; P_{\tilde{Y}_0 | \tilde{X}^l_{-\delta}} | P_{X^l_{t-\delta}}) = \sum_{x \in \mathcal{X}^l} P_{X^l_{t-\delta}}(x) \mathbb{E}_{Y \sim P_{Y_t | X^l_{t-\delta} = x}} \left[L\left(Y, a_{P_{\tilde{Y}_0 | \tilde{X}^l_{-\delta} = x}}\right) \right], \tag{5.7}$$

where the L-conditional cross entropy $H_L(P_{Y|X}; P_{\tilde{Y}|\tilde{X}} | P_X)$ is defined as [12]

$$H_L(P_{Y|X}; P_{\tilde{Y}|\tilde{X}} | P_X) = \sum_{x \in \mathcal{X}} P_X(x) \mathbb{E}_{Y \sim P_{Y|X=x}} \left[L\left(Y, a_{P_{\tilde{Y}|\tilde{X}=x}}\right) \right]. \tag{5.8}$$

If training algorithm considers sets of large and wide neural networks such that $a_{P_{\tilde{Y}_0 | \tilde{X}^l_{-\delta} = x}}$ and $\phi_l(\delta, x)$ for all $\delta \in \mathbb{Z}^+$ and $x \in \mathcal{X}^l$ are close to each other, then the difference between the inference error $\text{err}_{\text{inference}}(\delta, l)$ and the L-conditional cross entropy $H_L(P_{Y_t | X^l_{t-\delta}}; P_{\tilde{Y}_0 | \tilde{X}^l_{-\delta}} | P_{X^l_{t-\delta}})$ is small [12]. Compared to $\text{err}_{\text{inference}}(\delta, l)$, the L-conditional cross entropy $H_L(P_{Y_t | X^l_{t-\delta}}; P_{\tilde{Y}_0 | \tilde{X}^l_{-\delta}} | P_{X^l_{t-\delta}})$ are mathematically more convenient to analyze, as we will see next.

Information-theoretic Monotonicity Analysis

The following lemma interprets the monotonicity of the L-conditional entropy $H_L(\tilde{Y}_0 | \tilde{X}^l_{-\delta})$ and the L-conditional cross entropy $H_L(P_{Y_t | X^l_{t-\delta}}; P_{\tilde{Y}_0 | \tilde{X}^l_{-\delta}} | P_{X^l_{t-\delta}})$ with respect to the feature length l.

Lemma 5.1 *The following assertions are true:*

(a) Given $\delta \geq 0$, $H_L(\tilde{Y}_0|\tilde{X}^l_{-\delta})$ is a non-increasing function of l, i.e., for all $1 \leq l_1 \leq l_2$

$$H_L(\tilde{Y}_0|\tilde{X}^{l_2}_{-\delta}) \leq H_L(\tilde{Y}_0|\tilde{X}^{l_1}_{-\delta}). \tag{5.9}$$

(b) Given $\beta \geq 0$, if for all $l = 1, 2, \ldots,$ and $x \in \mathcal{V}^l$

$$\sum_{x \in \mathcal{X}^l} P_{X^l_{t-\delta}}(x) \sum_{y \in \mathcal{Y}} (P_{Y_t|X^l_{t-\delta}=x}(y) - P_{\tilde{Y}_0|\tilde{X}^l_{-\delta}=x}(y))^2 \leq \beta^2, \tag{5.10}$$

then for all $1 \leq l_1 \leq l_2$

$$H_L(P_{Y_t|X^{l_2}_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^{l_2}_{-\delta}}|P_{X^{l_2}_{t-\delta}}) \leq H_L(P_{Y_t|X^{l_1}_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^{l_1}_{-\delta}}|P_{X^{l_1}_{t-\delta}}) + O(\beta). \tag{5.11}$$

Proof 5.1 *Lemma 1 can be proven by using the data processing inequality for L-conditional entropy [48, Lemma 12.1] and a local information geometric analysis. See Appendix 5.B for the details.*

Lemma 5.1(a) demonstrates that for a given AoI value δ, the L-conditional entropy $H_L(\tilde{Y}_0|\tilde{X}^l_{-\delta})$ decreases as the feature length l increases. This is due to the fact that a longer feature provides more information, consequently leading to a lower L-conditional entropy. Additionally, as indicated in Lemma 5.1(b), when the conditional distributions in training and inference data are close to each other (i.e., when β in (5.10) is close to 0), the L-conditional cross entropy $H_L(P_{Y_t|X^l_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^l_{-\delta}}|P_{X^l_{t-\delta}})$ is close to a non-increasing function of the feature length l. This information-theoretic analysis clarifies the experimental results depicted in Fig. 2.5(a) and Fig. 5.3(a), where the inference error diminishes with the increasing feature length.

5.4 Learning and Communications Co-design: Single Source Case

Let $d(t)$ denote the feature length of the most recently received feature in time slot t. The time-averaged expected inference error under policy $\pi = ((S_1, b_1, l_1), (S_2, b_2, l_2), \ldots)$ is

expressed as

$$\bar{p}_{\pi} = \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\pi}\left[\sum_{t=0}^{T-1} \mathrm{err}_{\mathrm{inference}}(\Delta(t), d(t))\right], \tag{5.12}$$

where $\bar{p}_{\pi}$ is denoted as the time-averaged inference error, and $\mathrm{err}_{\mathrm{inference}}(\Delta(t), d(t))$ is the expected inference error at time t corresponding to the system state $(\Delta(t), d(t))$. In this section, we slove two problems. The first one is to find an optimal policy that minimizes the time-averaged expected inference error among all the causal policies in Π_{inv} that consider time-invariant feature length. Another problem is to find an optimal policy that minimizes the time-averaged expected inference error among all the causal policies in Π.

5.4.1 Time-invariant Feature Length

We first find an optimal policy that minimizes the time-averaged inference error among all causal policies with time-invariant feature length in Π_{inv} defined in (5.4):

$$\bar{p}_{\mathrm{inv}} = \inf_{\pi\in\Pi_{\mathrm{inv}}} \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\pi}\left[\sum_{t=0}^{T-1} \mathrm{err}_{\mathrm{inference}}(\Delta(t), d(t))\right], \tag{5.13}$$

where $\bar{p}_{\mathrm{inv}}$ is the optimum value of (5.13). The problem (5.13) is an infinite time-horizon average-cost semi-Markov decision process (SMDP). Such problems are often challenging to solve analytically or with closed-form solutions. The per-slot cost function $\mathrm{err}_{\mathrm{inference}}(\Delta(t), d(t))$ in (5.13) depends on two variables: the AoI $\Delta(t)$ and the feature length $d(t)$. Prior studies [21,23,25–27,29,32,45,79] have considered linear and non-linear monotonic AoI functions. Due to the fact that (i) the cost function in (5.13) depends on two variables and (ii) is not necessarily monotonic with respect to AoI, finding an optimal solution is challenging and the existing scheduling policies cannot be directly applied to solve (5.13). Therefore, it is necessary to develop a new scheduling policy that can address the complexities of (5.13).

Surprisingly, we get a closed-form solution of (5.13). To present the solution, we define a function $\gamma_l(\delta, d)$ as

$$\gamma_l(\delta, d) := \inf_{\tau \in \{1,2,\ldots\}} \frac{1}{\tau} \sum_{j=0}^{\tau-1} \mathbb{E}\Big[\mathrm{err}_{\mathrm{inference}}\Big(\delta + j + T_1(l), d\Big)\Big]. \tag{5.14}$$

Theorem 5.1 *If $T_i(l)$'s are i.i.d. with a finite mean $\mathbb{E}[T_i(l)]$ for each $l = 1, 2, \ldots, B$, then there exists an optimal solution $\pi^* = ((S_1^*, b_1^*, l^*), (S_2^*, b_2^*, l^*), \ldots) \in \Pi_{\mathrm{inv}}$ to (5.13) that satisfies:*

(a) The optimal feature position in π^* is time-invariant, i.e., $b_1^* = b_2^* = \cdots = b^*$. The optimal feature length l^* and the optimal feature position b^* in π^* are given by

$$(l^*, b^*) = \operatorname*{argmin}_{\substack{l \in \mathbb{Z}, b \in \mathbb{Z} \\ 1 \le l \le B, 0 \le b \le B - l}} \beta_{b,l}, \tag{5.15}$$

where $\beta_{b,l}$ is the unique root of equation

$$\mathbb{E}\left[\sum_{t=D_i(\beta_{b,l})}^{D_{i+1}(\beta_{b,l})-1} \mathrm{err}_{\mathrm{inference}}(\Delta_b(t), l)\right] - \beta_{b,l}\ \mathbb{E}\Big[D_{i+1}(\beta_{b,l}) - D_i(\beta_{b,l})\Big] = 0, \tag{5.16}$$

$D_i(\beta_{b,l}) = S_i(\beta_{b,l}) + T_i(l)$, $\Delta_b(t) = t - S_i(\beta_{b,l}) + b$, the sequence $(S_1(\beta_{b,l}), S_2(\beta_{b,l}), \ldots)$ is determined by

$$S_{i+1}(\beta_{b,l}) = \min_{t \in \mathbb{Z}} \big\{ t \ge D_i(\beta_{b,l}) : \gamma_l(\Delta_b(t), l) \ge \beta_{b,l} \big\}, \tag{5.17}$$

and the function $\gamma_l(\cdot)$ is defined in (5.14).

(b) The optimal scheduling time S_{i+1}^* in π^* is determined by

$$S_{i+1}^* = \min_{t \in \mathbb{Z}} \big\{ t \ge S_i^* + T_i(l^*) : \gamma_{l^*}(\Delta_{b^*}(t), l^*) \ge \bar{p}_{\mathrm{inv}} \big\}, \tag{5.18}$$

where $\Delta_{b^*}(t) = t - S_i^* + b^*$ *is the AoI at time* t*. The optimal objective value* $\bar{p}_{\text{inv}}$ *of* (5.13) *is*

$$\bar{p}_{\text{inv}} = \min_{\substack{l\in\mathbb{Z}, b\in\mathbb{Z} \\ 1\le l\le B, 0\le b\le B-l}} \beta_{b,l}. \tag{5.19}$$

We prove Theorem 5.1 in two steps: (i) We find B policies, each of which is optimal among the set of policies Π_l where $l = 1, 2, \ldots, B$. After that (ii) we select the policy that results in the minimum average inference error among the B policies. See Appendix 5.C for details.

Theorem 5.1 implies that the optimal scheduling policy has a nice structure. According to Theorem 5.1(a), the feature position b_i^* is constant for all i-th features, i.e., $b_1^* = b_2^* = \ldots = b^*$. The optimal feature length l^* and the optimal feature position b^* are pre-computed by solving (5.15) and then used in real-time. The parameter $\beta_{b,l}$ in (5.15) is the unique root of (5.16), which is solved by using low-complexity algorithms, e.g., bisection search, newtons method, and fixed point iteration [25]. Theorem 5.1(b) implies that the optimal scheduling time S_{i+1}^* follows a threshold policy. Specifically, a feature is transmitted in time-slot t if the following two conditions are satisfied: (i) The channel is idle in time-slot t and (ii) the value $\gamma_{l^*}(\Delta(t), l^*)$ exceeds the optimal objective value $\bar{p}_{\text{inv}}$ of (5.13). The optimal objective value $\bar{p}_{\text{inv}}$ is obtained from (5.19). Our threshold-based scheduling policy has a significant distinction from previous threshold-based policies studied in the literature, such as [23, 25, 27, 45]. In these prior works, the threshold function used to determine the scheduling time is based solely on the AoI value and is non-decreasing with respect to AoI. However, in our proposed strategy, (i) the threshold function $\gamma_l(\cdot)$ depends on both the AoI value and the feature length and (ii) the threshold function $\gamma_l(\cdot)$ can be non-monotonic with respect to AoI.

Monotonic AoI Cost function

Consider a special case where the inference error $\text{err}_{\text{inference}}(\delta, l)$ is a non-decreasing function of δ for every feature length l. A simplified solution can be derived for this specific case of (5.13). In this scenario, the optimal feature position is $b^* = 0$, and the threshold

function $\gamma_l(\cdot)$ defined in (5.14) becomes:

$$\gamma_l(\delta, d) = \mathbb{E}\left[\text{err}_{\text{inference}}\Big(\delta + T_1(l), d\Big)\right]. \tag{5.20}$$

In this special case of monotonic AoI cost function, (5.18) can be rewritten as a threshold policy of the AoI $\Delta(t)$ in the form of $\Delta(t) \geq w(l^*, \bar{p}_{\text{inv}})$, where $w(l, \beta)$ is defined as:

$$w(l, \beta) = \inf\left\{\delta \geq 0 : \mathbb{E}\left[\text{err}_{\text{inference}}\Big(\delta + T_1(l), l\Big)\right] \geq \beta\right\}. \tag{5.21}$$

However, when $\text{err}_{\text{inference}}(\delta, l)$ is not monotonic with respect to AoI δ, (5.18) cannot be reformulated as a threshold policy of the AoI $\Delta(t)$. This is a key difference with earlier studies [23, 26, 27].

Connection with Restart-in-state Problem

Consider another special case in which all features take 1 time-slot for transmission. For this special case, the threshold function $\gamma_l(\cdot)$ defined in (5.14) becomes

$$\gamma_l(\delta, d) = \inf_{\tau \in \{1,2,\ldots\}} \frac{1}{\tau} \sum_{j=0}^{\tau-1} \mathbb{E}\Big[\text{err}_{\text{inference}}\Big(\delta + j + 1, d\Big)\Big]. \tag{5.22}$$

This special case of (5.13) is a restart-in-state problem [70, Chapter 2.6.4]. This is because whenever a feature with the optimal feature length l^* and from the optimal feature position b^* is transmitted, AoI value restarts from b^*+1 in the next time slot. For this restart-in-state problem, the optimal sending time follows a threshold policy [70, Chapter 2.6.4]. Specifically, a feature is transmitted if

$$h(\Delta_{b^*}(t+1), l^*) \geq h(b^* + 1, l^*), \tag{5.23}$$

where the relative value function $h(\delta, l^*)$ of the restart-in-state problem is given by

$$h(\delta, l^*) = \min_{Z \in \{0,1,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{Z}\Big(\text{err}_{\text{inference}}(\delta + k, l^*) - \bar{p}_{\text{inv}}\Big)\right] + h(b^* + 1, l^*). \tag{5.24}$$

By using (5.24), we can show that (5.23) is equivalent to

$$\gamma_{l^*}(\Delta_{b^*}(t), l^*) \geq \bar{p}_{\text{inv}}. \tag{5.25}$$

where the function $\gamma_l(\delta, d)$ is defined in (5.22). This connection between the restart-in-state problem and AoI minimization was unknown before. The original problem considers more general $T_i(l)$, which can be considered as a restart-in-random state problem. This is because whenever i-th feature with optimal feature length l^* and from optimal feature position b^* is transmitted, AoI restarts from a random value $b^* + T_i(l^*)$ after $T_i(l^*)$ time slots.

5.4.2 Time-variant Feature Length

Now, we find an optimal scheduling policy that minimizes time-averaged inference error among all causal policies in Π:

$$\bar{p}_{opt} = \inf_{\pi \in \Pi} \limsup_{T \to \infty} \frac{1}{T} \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \text{err}_{\text{inference}}(\Delta(t), d(t)) \right], \tag{5.26}$$

where $\text{err}_{\text{inference}}(\Delta(t), d(t))$ is the inference error at time slot t and $\bar{p}_{opt}$ is the optimum value of (5.26). Because $\Pi_{\text{inv}} \subset \Pi$,

$$\bar{p}_{opt} \leq \bar{p}_{\text{inv}}, \tag{5.27}$$

where $\bar{p}_{\text{inv}}$ is the optimum value of (5.13). Like (5.13), problem (3.3) can also be expressed as an infinite time-horizon average-cost SMDP. Note that (3.3) is more complex SMDP than (5.13) because the feature length in (5.26) is allowed to vary over time.

The optimal policy can be determined by using a dynamic programming method associated with the average cost SMDP [69, 72]. There exists a function $h(\cdot)$ such that for all $\delta \in \mathbb{Z}^+$ and $0 \leq d \leq B$, the optimal objective value $\bar{p}_{opt}$ of (5.26) satisfies the following

Bellman equation:

$$h(\delta,d)=\min_{\substack{Z\in\{0,1,\ldots\}\\ l\in\mathbb{Z}:1\le l\le B\\ b\in\mathbb{Z}:0\le b\le B-l}}\mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{opt}\right)\right]+\mathbb{E}[h(T_1(l)+b,l)]. \quad (5.28)$$

Let $(Z^*(\delta,d),l^*(\delta,d),b^*(\delta,d))$ be the optimal solution to the Bellman equation (5.28). There exists an optimal solution $\pi^*=((S_1^*,b_1^*,l_1^*),(S_2^*,b_2^*,l_2^*),\ldots)\in\Pi$ to (3.3), determined by

$$l_{i+1}^*=l^*(T_i(l_i^*)+b_i^*,l_i^*), \quad (5.29)$$

$$b_{i+1}^*=b^*(T_i(l_i^*)+b_i^*,l_i^*), \quad (5.30)$$

$$S_{i+1}^*=S_i^*+T_i(l_i^*)+Z^*(T_i(l_i^*)+b_i^*,l_i^*), \quad (5.31)$$

where $Z^*(T_i(l_i^*)+b_i^*,l_i^*)$ is the optimal waiting time for sending the $(i+1)$-th feature after the i-th feature is delivered.

To get the optimal policy π^*, we need to solve (5.28). Solving (5.28) is complex as it requires joint optimization of three variables. Moreover, an optimal solution obtained by the dynamic programming method provides no insight. We are able to simplify (5.28) in Theorem 5.2 by analyzing the structure of the optimal solution.

Theorem 5.2 *The following assertions are true:*

(a) If $T_i(l)$'s are i.i.d. with a finite mean $\mathbb{E}[T_i(l)]$ for each $l=1,2,\ldots,B$, then there exists a function $h(\cdot)$ such that for all $\delta\in\mathbb{Z}^+$ and $0\le d\le B$, the optimal objective value $\bar{p}_{opt}$ of (5.26) satisfies the following Bellman equation:

$$\begin{aligned}&h(\delta,d)\\ &=\min_{\substack{l\in\mathbb{Z}\\ 1\le l\le B}}\left\{\mathbb{E}\left[\sum_{k=0}^{Z_l(\delta,d)+T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{opt}\right)\right]+\min_{\substack{b\in\mathbb{Z}\\ 0\le b\le B-l}}\mathbb{E}[h(T_1(l)+b,l)]\right\},\end{aligned} \quad (5.32)$$

where $h(\cdot)$ is called the relative value function and the function $Z_l(\delta, d)$ is given by

$$Z_l(\delta, d) = \min_{\tau \in \mathbb{Z}} \{\tau \geq 0 : \gamma_l(\delta + \tau, d) \geq \bar{p}_{opt}\}, \tag{5.33}$$

and the function $\gamma_l(\delta, d)$ is defined in (5.14).

(b) In addition, there exists an optimal solution $\pi^* = ((S_1^*, b_1^*, l_1^*), (S_2^*, b_2^*, l_2^*), \ldots) \in \Pi$ to (5.26) *that is determined by*

$$\begin{aligned} l_{i+1}^* = &\underset{\substack{l \in \mathbb{Z} \\ 1 \leq l \leq B}}{\operatorname{argmin}} \left\{ \mathbb{E}\left[\sum_{k=0}^{\substack{Z_l(T_1(l_i^*)+b_i^*, l_i^*) \\ +T_1(l)-1}} \left(\mathrm{err}_{\mathrm{inference}}(\Delta(D_i)+k, l_i^*) - \bar{p}_{opt} \right) \right] \right. \\ &\left. + \min_{\substack{b \in \mathbb{Z} \\ 0 \leq b \leq B-l}} \mathbb{E}[h(T_1(l)+b, l)] \right\}, \end{aligned} \tag{5.34}$$

$$b_{i+1}^* = \underset{b \in \mathbb{Z}: 0 \leq b \leq B - l_{i+1}^*}{\operatorname{argmin}} \mathbb{E}[h(T_1(l_{i+1}^*) + b, l_{i+1}^*)], \tag{5.35}$$

$$S_{i+1}^* = \min_{t \in \mathbb{Z}} \{t \geq D_i : \gamma_{l_{i+1}^*}(\Delta(t), l_i^*) \geq \bar{p}_{opt}\}, \tag{5.36}$$

where $\Delta(t) = t - S_i^* + b_i^*$ is the AoI at time t and $D_i = S_i^* + T_i(l_i^*)$ is the i-th feature delivery time.

Theorem 5.2(a) simplifies the Bellman equation (5.28) to (3.22). Unlike (5.28), which involves joint optimization of three variables, (5.32) is an integer optimization problem. This simplification is possible because, for a given feature length l, the original equation (5.28) can be separated into two separated optimization problems. The first problem involves finding the optimal stopping time, denoted by $Z_l(\delta, d)$ defined in (5.33), and the second problem is to determine the feature position b that minimizes $\mathbb{E}[h(T_1(l) + b, l)]$. By breaking down the original equation in this way, we can solve the problem more efficiently. Detailed proof of Theorem 5.2 can be found in Appendix 5.D.

Furthermore, Theorem 5.2(a) provides additional insights into the solution of (5.28). Theorem 5.2(a) implies that the optimal stopping time $Z^*(\delta, d)$ in (5.28) follows a threshold policy. Specifically, if $l^*(\delta, d) = l$, then $Z^*(\delta, d)$ equals $Z_l(\delta, d)$, which is defined in (5.33).

Algorithm 2 Policy Evaluation Algorithm

Input: $Z_\pi(\delta, d)$, $l_\pi(\delta, d)$, and $b_\pi(\delta, d)$ for all (δ, d).
Initialize $h_\pi(\delta, d)$ arbitrarily for all (δ, d), except for one fixed state (δ', d') with $h_\pi(\delta', d') = 0$.
Initialize a small positive number α_1 as a threshold.
repeat
    $\theta_1 \leftarrow 0$.
    Determine $\bar{p}_\pi$ using (5.37).
    for each state (δ, d) **do**
        $\tau' \leftarrow Z_\pi(\delta, d) + T_1(l_\pi(\delta, d))$.
        $h'_\pi(\delta, d) \leftarrow \mathbb{E}[\sum_{k=0}^{\tau'-1}(\text{err}_{\text{inference}}(\delta+k, d)-\bar{p}_\pi)]+\mathbb{E}[h_\pi(T_1(l_\pi(\delta, d))+b_\pi(\delta, d), l_\pi(\delta, d))]$.
        $\theta_1 \leftarrow \max\{\theta_1, \left|h'_\pi(\delta, d) - h_\pi(\delta, d)\right|\}$.
    end for
    $h_\pi \leftarrow h'_\pi$.
until $\theta_1 \leq \alpha_1$.
return $\bar{p}_\pi$ and $h_\pi(\cdot)$.

Here, $Z_l(\delta, d)$ is the minimum positive integer value τ for which $\gamma_l(\delta + \tau, d)$ defined in (5.14) exceeds the optimal objective value $\bar{p}_{opt}$.

Theorem 5.2(b) provides an optimal solution $\pi^* \in \Pi$ to (5.26). According to Theorem 5.2(b), by using precomputed $\bar{p}_{opt}$ and the relative value function $h(\cdot)$, we can obtain the optimal feature length l^*_{i+1} from (5.34) using an exhaustive search algorithm. After obtaining l^*_{i+1}, the optimal feature position b^*_{i+1} can be determined from (5.35). The optimal scheduling time S^*_{i+1} provided in (5.36) follows a threshold policy. Specifically, the $(i + 1)$-th feature is transmitted in time-slot t if two conditions are satisfied: (i) the previous feature is delivered by time t, and (ii) the function $\gamma_{l^*_{i+1}}(\Delta(t), l^*_i)$ exceeds the optimal objective value $\bar{p}_{opt}$ of (5.26).

Policy Iteration Algorithm for Computing $\bar{p}_{opt}$ and $h(\cdot)$

To effectively implement the optimal solution $\pi^* \in \Pi$ for (5.26), as outlined in Theorem 5.2, it is necessary to precompute the optimal objective value $\bar{p}_{opt}$ and the relative value function $h(\cdot)$ that satisfies the Bellman equation (5.32). The computation of $\bar{p}_{opt}$ and $h(\cdot)$ can be achieved by employing policy iteration algorithm or relative value iteration algorithm for SMDPs, as detailed in [72, Section 11.4.4]. To apply the relative value iteration algorithm, we need to transform the SMDP into an equivalent MDP. However, this transformation process

can be challenging to execute. Therefore, in this paper, we opt to utilize the policy iteration algorithm specifically tailored for SMDPs [72, Section 11.4.4]. Algorithm 3 provides a policy iteration algorithm for obtaining $\bar{p}_{opt}$ and $h(\cdot)$, which is composed of two steps: (i) *policy evaluation* and (ii) *policy improvement*.

Policy Evaluation: Let h_π and $\bar{p}_\pi$ be the relative value function and the average inference error under policy π. Let $l_\pi(\delta, d)$, $b_\pi(\delta, d)$, and $Z_\pi(\delta, d)$ represent feature length, feature position, and waiting time for sending the $(i+1)$-th feature under policy π when $\Delta(D_i) = \delta$ and $d(D_i) = d$. Given $l_\pi(\delta, d)$, $b_\pi(\delta, d)$, and $Z_\pi(\delta, d)$ for all (δ, d), we can evaluate the relative value function $h_\pi(\cdot)$ and the average inference error $\bar{p}_\pi$ using Algorithm 2. The relative value function $h_\pi(\delta, d)$ represents relative value associated with a reference state. We can set (δ', d') as a reference state with $h_\pi(\delta', d') = 0$. By using $h_\pi(\delta', d') = 0$, the average inference error $\bar{p}_\pi$ is determined by

$$\bar{p}_\pi = \frac{1}{\mathbb{E}[\tau]} \left(\mathbb{E}\left[\sum_{k=0}^{\tau-1} \mathrm{err}_{\mathrm{inference}}(\delta' + k, d') \right] + \mathbb{E}[h_\pi(T_1(l_\pi(\delta', d')) + b_\pi(\delta', d'), l_\pi(\delta', d'))] \right), \tag{5.37}$$

where $\tau = Z_\pi(\delta', d') + T_1(l_\pi(\delta', d'))$. We then use an iterative procedure within Algorithm 2 to determine the relative value function $h_\pi(\cdot)$.

Policy Improvement: After obtaining h_π and $\bar{p}_\pi$ from Algorithm 2, we apply Theorem 5.2 to derive an improved policy π' in Algorithm 3. Feature length $l_{\pi'}(\delta, d)$, feature position $b_{\pi'}(\delta, d)$, and waiting time $Z_{\pi'}(\delta, d)$ under policy π' is determined by

$$\begin{aligned} l_{\pi'}(\delta, d) = \underset{1 \le l \le B}{\mathrm{argmin}} \Bigg\{ & \mathbb{E}\left[\sum_{k=0}^{Z_l(\delta,d)+T_1(l)-1} \mathrm{err}_{\mathrm{inference}}(\delta + k, d) \right] \\ & - \mathbb{E}[Z_l(\delta, d) + T_1(l)] \bar{p}_\pi \\ & + \min_{0 \le b \le B-l} \mathbb{E}[h_\pi(T_1(l) + b, l)] \Bigg\}, \end{aligned} \tag{5.38}$$

$$b_{\pi'}(\delta, d) = \underset{0 \le b \le B - l_{\pi'}(\delta,d)}{\mathrm{argmin}} \mathbb{E}[h_\pi(T_1(l_{\pi'}(\delta, d)) + b, l_{\pi'}(\delta, d))], \tag{5.39}$$

Algorithm 3 Policy Iteration Algorithm

Initialize $Z_\pi(\delta, d)$, $l_\pi(\delta, d)$, and $b_\pi(\delta, d)$ for all (δ, d).
Initialize a small positive number α_2 as threshold.
repeat
    $\theta_2 \leftarrow 0$.
    Obtain $h_\pi(\cdot)$ and $\bar{p}_\pi$ from Algorithm 2.
    for all (δ, d) **do**
        Get $l_{\pi'}(\delta, d)$, $b_{\pi'}(\delta, d)$, $Z_{\pi'}(\delta, d)$ using (5.38)-(5.40).
        $\theta_2 \leftarrow \max\Big\{\theta_2, |l_{\pi'}(\delta, d) - l_\pi(\delta, d)|$
            $+ |b_{\pi'}(\delta, d) - b_\pi(\delta, d)| + |Z_{\pi'}(\delta, d) - Z_\pi(\delta, d)|\Big\}$.
        $l_\pi(\delta, d) \leftarrow l_{\pi'}(\delta, d)$.
        $b_\pi(\delta, d) \leftarrow b_{\pi'}(\delta, d)$.
        $Z_\pi(\delta, d) \leftarrow Z_{\pi'}(\delta, d)$.
    end for
until $\theta_2 \leq \alpha_2$.
return $\bar{p}_{opt} \leftarrow \bar{p}_\pi$ and $h \leftarrow h_\pi$.

$$Z_{\pi'}(\delta, d) = \min_{\tau \in \{0,1,\ldots\}} \big\{\tau \geq 0 : \gamma_{l_{\pi'}(\delta,d)}(\delta + \tau, d) \geq \bar{p}_\pi\big\}. \tag{5.40}$$

Instead of a joint optimization problem (5.28), Algorithm 3 utilizes separated optimization problems (5.38)-(5.40) based on Theorem 5.2. If the improved policy π' is equal to the old policy π, then the policy iteration algorithm converges. Theorem 11.4.6 in [72] establishes the finite convergence of the policy iteration algorithm of an average cost SMDP.

Now, we discuss the time-complexity of Algorithms 2-3. To manage the infinite set of AoI values in practice, we introduce an upper bound denoted as δ_{bound}. Whenever δ exceeds δ_{bound}, we set $h_\pi(\delta, d) = h_\pi(\delta_{\text{bound}}, d)$ for all d. Hence, each iteration of our policy evaluation step requires one pass through the approximated state space $\{1, 2, \ldots, \delta_{\text{bound}}\} \times \{1, 2, \ldots, B\}$. Therefore, the time complexity of each iteration is $O(\delta_{\text{bound}} B)$, assuming that the required expected values are precomputed. Considering the bounded set $\{0, 1, \ldots, \delta_{\text{bound}}\}$ instead of $\mathbb{Z}^+$, the time complexities of (5.38), (5.39), and (5.40) are $O(B^2)$, $O(B)$, and $O(\delta_{\text{bound}})$, respectively, provided that the expected values in (5.38)-(5.40) are precomputed. The overall complexity of (5.38)-(5.40) is $O(\max\{B^2, B, \delta_{\text{bound}}\})$, which is more efficient than the joint optimization problem (5.28). The latter has a time complexity of $O(\delta_{\text{bound}} B^2)$. In each iteration of the policy improvement step, the optimization problems (5.38)-(5.40) are solved

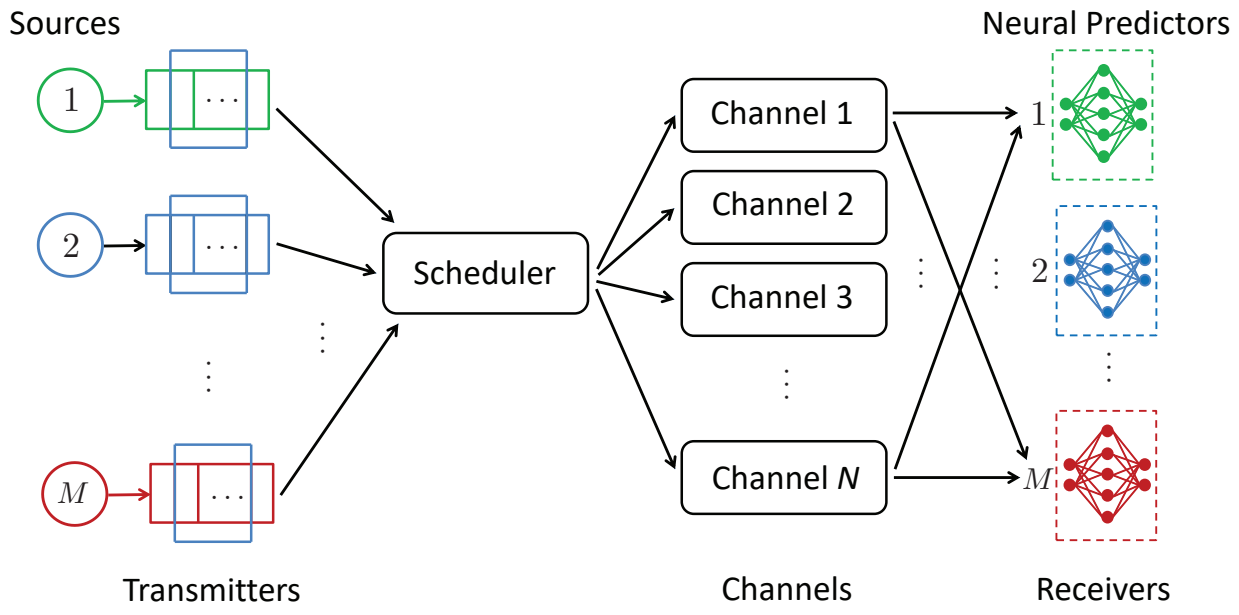


Figure 5.4: A multiple source-predictor pairs and multiple channel remote inference system.

for all state (δ, d) such that $\delta = 1, 2, \ldots, \delta_{\text{bound}}$ and $d = 1, 2, \ldots, B$. Hence, the total complexity of each iteration of the policy improvement step is $O(\max\{B^3\delta_{\text{bound}}, B\delta_{\text{bound}}^2\})$.

5.5 Learning and Communications Co-design: Multiple Source Case

System Model

Consider a remote inference system consisting of $M \geq 1$ source-predictor pairs connected through $N \geq 1$ shared communication channels, as illustrated in Fig. 5.4. Each source j has a buffer that stores B_j most recent signal observations at each time slot t. At time slot t, a centralized scheduler determines whether to send a feature from source j with feature length $l_j(t)$ and feature position $b_j(t)$. We denote $l_j(t) = 0$ if the scheduler decides not to send a feature from source j at time t. If a feature from source j is sent, we assume it will be delivered to the j-th neural predictor in the next time slot using $l_j(t)$ channel resources. The transmission model of the multiple source system is significantly different from that of the single source model discussed in Chapter 5.2. In the latter case, only one channel was considered, while N communication channels are available in the former. The channels

could be from multiple frequencies and/or time resources. For example, if the clock rate in the multiple access control (MAC) layer is faster than that of the application layer, then one application-layer time-slot could comprise multiple MAC-layer time-slots. A feature can utilize multiple channels (i.e., frequency or time resources) for transmission during a single time slot. However, the channel resource is limited, so the system must satisfy

$$\sum_{j=1}^{M} l_j(t) \leq N. \tag{5.41}$$

The system begins operating at time $t = 0$. Let $S_{j,i}$ denote the sending time of the i-th feature from the j-th source. Since we assume that a feature takes one time-slot to transmit, the corresponding neural predictor receives the i-th feature from the j-th source at time $S_{j,i} + 1$. The AoI of the source j at time slot t is defined as

$$\Delta_j(t) := t - S_{j,i} + b_j(S_{j,i}), \text{ if } S_{j,i} < t \leq S_{j,i+1}. \tag{5.42}$$

We denote $d_j(t)$ as the feature length of the most recent received feature from j-th source by time t, given by

$$d_j(t) = l_j(S_{j,i}), \text{ if } S_{j,i} < t \leq S_{j,i+1}. \tag{5.43}$$

Scheduling Policy

At time slot t, a centralized scheduler determines the value of the feature length $l_j(t)$ and the feature position $b_j(t)$ for every j-th source. A scheduling policy is denoted by $\pi = (\pi_j)_{j=1}^{M}$, where $\pi_j = ((l_j(1), b_j(1)), (l_j(2), b_j(2)), \ldots)$. Let Π denote the set of all the causal scheduling policies that determine $l_j(t)$ and $b_j(t)$ based on the current and the historical information available at the transmitter such that $0 \leq l_j(t) + b_j(t) \leq B_j$.

Problem Formulation

Our goal is to minimize the time-averaged sum of the inference errors of the M sources, which is formulated as

$$\inf_{\pi\in\Pi} \sum_{j=1}^{M} \limsup_{T\to\infty} \mathbb{E}_{\pi} \left[\frac{1}{T} \sum_{t=0}^{T-1} p_j(\Delta_j(t), d_j(t)) \right], \tag{5.44}$$

$$\text{s.t.} \sum_{j=1}^{M} l_j(t) \leq N, \quad t = 0, 1, 2, \ldots, \tag{5.45}$$

where $p_j(\Delta_j(t), d_j(t))$ is the inference error of source j at time slot t.

The problem (5.44)-(5.45) can be cast into an infinite-horizon average cost restless multi-armed multi-action bandit problem [33, 89] by viewing each source j as an arm, where a scheduler needs to decide multiple actions $(l_j(t), b_j(t))_{j=1}^{M}$ at every time t by observing state $(\Delta_j(t), d_j(t))$.

Finding an optimal solution to the RMAB problem is PSPACE hard [75]. Whittle, in his seminal work [33], proposed a heuristic policy for RMAB problem with binary action. In [89], a modified Whittle index policy is proposed for the multi-action RMAB problems. Whittle index policy is known to be asymptotically optimal [90], but the policy needs to satisfy a complicated indexability condition. Proving indexability is challenging for our multi-action RMAB problem because we allow (i) general penalty function $p_j(\delta, l)$ that is not necessarily monotonic with respect to AoI δ and (ii) time-variant feature length. To this end, we propose a low-complexity algorithm that does not need to satisfy any indexability condition.

Lagrangian Optimization of a Relaxed Problem

Similar to Whittle's approach [33], we utilize a Lagrange relaxation of the problem (5.44)-(5.45). We first relax the per time-slot channel constraint (5.45) as the following time-average expected channel constraint

$$\sum_{j=1}^{M} \limsup_{T\to\infty} \mathbb{E}_{\pi} \left[\frac{1}{T} \sum_{t=0}^{T-1} l_j(t) \right] \leq N. \tag{5.46}$$

The relaxed constraint (5.46) only needs to be satisfied on average, whereas (5.45) is required to hold at every time-slot. By this, the original problem (5.44)-(5.45) becomes

$$\inf_{\pi\in\Pi}\sum_{j=1}^{M}\limsup_{T\to\infty}\mathbb{E}_{\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1}p_j(\Delta_j(t),d_j(t))\right], \tag{5.47}$$

$$\text{s.t.}\sum_{j=1}^{M}\limsup_{T\to\infty}\mathbb{E}_{\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1}l_j(t)\right]\leq N. \tag{5.48}$$

The relaxed problem (5.47)-(5.48) is of interest as the optimal solution of the problem provides a lower bound to the original problem (5.44)-(5.45).

Lagrangian Dual Decomposition of (5.47)**-**(5.48)

To solve (5.47)-(5.48), we utilize a Lagrangian dual decomposition method [33, 76]. At first, we apply Lagrangian multiplier $\lambda \geq 0$ to the time-average channel constraint (5.48) and get the following Lagrangian dual function

$$q(\lambda)=\inf_{\pi\in\Pi}\sum_{j=1}^{M}\limsup_{T\to\infty}\mathbb{E}_{\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1}\Big(p_j(\Delta_j(t),d_j(t))+\lambda l_j(t)\Big)\right]-\lambda N. \tag{5.49}$$

The problem (5.49) can be decomposed into M sub-problems. The sub-problem associated with the j-th source is defined as:

$$\bar{p}_j(\lambda)=\inf_{\pi_j\in\Pi_j}\limsup_{T\to\infty}\frac{1}{T}\mathbb{E}_{\pi_j}\left[\sum_{t=0}^{T-1}\Big(p_j(\Delta_j(t),d_j(t))+\lambda l_j(t)\Big)\right], \tag{5.50}$$

where Π_j is the set of all causal scheduling policies π_j. The sub-problem (5.50) is an infinite horizon average cost MDP, where a scheduler decides action $(l_j(t), b_j(t))$ by observing state $(\Delta_j(t), d_j(t))$. The Lagrange multiplier λ in (5.50) can be interpreted as a transmission cost: whenever $l_j(t) = l$, the source j has to pay cost of λl for using l channel resources.

The optimal solution to (5.50) can be obtained by solving the following Bellman equation:

$$h_{j,\lambda}(\delta, d) = \min_{\substack{l\in\mathbb{Z}, b\in\mathbb{Z} \\ 0\le l+b\le B_j}} Q_{j,\lambda}((\delta, d), (l, b)), \tag{5.51}$$

where $h_{j,\lambda}(\cdot)$ represents the relative value function of the MDP (5.50), and the function $Q_{j,\lambda}(\cdot,\cdot)$ is defined as follows

$$Q_{j,\lambda}((\delta, d), (l, b)) := \begin{cases} p_j(\delta, d) - \bar{p}_j(\lambda) + h_{j,\lambda}(\delta + 1, d), & \text{if } l = 0, \\ p_j(\delta, d) - \bar{p}_j(\lambda) + h_{j,\lambda}(b + 1, l) + \lambda l, & \text{otherwise.} \end{cases} \tag{5.52}$$

The relative value function $h_{j,\lambda}(\cdot)$ can be computed using the relative value iteration algorithm [69, 72].

Let $\pi^*_{j,\lambda} = ((l^*_{j,\lambda}(1), b^*_{j,\lambda}(1)), (l^*_{j,\lambda}(2), b^*_{j,\lambda}(2)), \ldots)$ be an optimal solution to (5.50), which is derived by using (5.51) and (5.52). The optimal feature length $l^*_{j,\lambda}(t)$ is determined by

$$l^*_{j,\lambda}(t) = \operatorname*{argmax}_{l\in\mathbb{Z}:0\le l\le B_j} h_{j,\lambda}(\Delta(t) + 1, d(t)) - h_{j,\lambda}(\hat{b}_{j,\lambda}(l) + 1, l) - \lambda l, \tag{5.53}$$

where the function $\hat{b}_{j,\lambda}(l)$ is given by

$$\hat{b}_{j,\lambda}(l) = \operatorname*{argmin}_{b\in\mathbb{Z}:0\le b\le B_j - l} h_{j,\lambda}(b + 1, l), \tag{5.54}$$

The optimal feature position in $\pi^*_{j,\lambda}$ is

$$b^*_{j,\lambda}(t) = \hat{b}_{j,\lambda}(l_{j,\lambda}(t)). \tag{5.55}$$

Algorithm 4 Dual Algorithm to Solve (5.56)

Input: Step size $\beta > 0$ and dual cost $\lambda(1) = 0$.
Initialize $\Delta_j(0)$, $d_j(0)$, $l_j(0)$, and $b_j(0)$ for all j.
Initialize a small positive number θ as threshold.
repeat
    for each source j **do**
        if $l_j(k-1) > 0$ **then**
            $\Delta_j(k) \leftarrow 1 + b_j(k-1)$, $d_j(k) \leftarrow l_j(k-1)$.
        else
            $\Delta_j(k) \leftarrow \Delta_j(k-1) + 1$, $d_j(k) \leftarrow d_j(k-1)$.
        end if
        Compute $l_{j,\lambda(k)}(k)$ using (5.53).
        Compute $b_{j,\lambda(k)}(k)$ using (5.55).
    end for
    Update $\lambda(k+1)$ using (5.57).
until $|\lambda(k+1) - \lambda(k)| \leq \theta$.
return $\lambda^* \leftarrow \lambda(k+1)$

Lagrange Dual Problem

Next, we determine the optimal dual cost λ^* that solves the following Lagrange dual problem:

$$\max_{\lambda \geq 0} q(\lambda), \tag{5.56}$$

where $q(\lambda)$ is the Lagrangian dual function defined in (5.49). To get λ^*, we apply the stochastic sub-gradient ascent method [76], which iteratively updates $\lambda(k)$ as follows

$$\lambda(k+1) = \max\left\{\lambda(k) + \frac{\beta}{k}\left(\sum_{j=1}^{M} l_{j,\lambda(k)}(k) - N\right), 0\right\}, \tag{5.57}$$

where k is the iteration index, $\beta > 0$ determines the step size $\frac{\beta}{k}$, and $l_{j,\lambda(k)}(k)$ is the feature length of source j at the k-th iteration. Detailed optimization technique is provided in Algorithm 4.

5.6 Net Gain Maximization Policy

After getting optimal dual cost λ^*, we can use policy $(\pi_{j,\lambda^*})_{j=1}^M$ for the relaxed problem (5.47)-(5.48). But it is infeasible to implement the policy for the original problem (5.44)-(5.45) because it may violate the scheduling constraint (5.45). Motivated by Whittle's approach [33], we aim to select actions with higher priority, while satisfying the scheduling constraint (5.45) at every time slot. Towards this end, we introduce "Net Gain", denoted as $\alpha_{j,\lambda}(\delta, d, l)$, to measure the advantage of selecting feature length l, which is given by

$$\alpha_{j,\lambda}(\delta, d, l) := Q_{j,\lambda}((\delta, d), (0, \hat{b}_{j,\lambda}(l))) - Q_{j,\lambda}((\delta, d), (l, \hat{b}_{j,\lambda}(l))), \tag{5.58}$$

where the function $Q_{j,\lambda}$ is defined in (5.52) and the function $\hat{b}_{j,\lambda}$ is defined in (5.54). Substituting (5.52) into (5.58), we get

$$\alpha_{j,\lambda}(\delta, d, l) = h_{j,\lambda}(\delta + 1, d) - h_{j,\lambda}(\hat{b}_{j,\lambda}(l) + 1, l) - \lambda l. \tag{5.59}$$

For a given λ, the net gain $\alpha_{j,\lambda}(\delta, d, l)$ has an economic interpretation. Given the state (δ, d) of source j, the net gain $\alpha_{j,\lambda}(\delta, d, l)$ measures the maximum reduction in the loss by selecting source j with feature length l, as opposed to not selecting source j at all. If $\alpha_{j,\lambda}(\delta, d, l)$ is negative for all $l = 1, 2, \ldots, B_j$, then it better not to select source j. If $\alpha_{j,\lambda}(\Delta_j(t), d_j(t), l_j) > \alpha_{k,\lambda}(\Delta_k(t), d_k(t), l_k)$, then the feature length l_j for source j is prioritized over the feature length l_k for source k. Under the constraint (5.45), we select feature lengths that maximize "Net Gain":

$$\max_{\substack{0 \le l_j(t) \le B \\ l_j(t) \in \mathbb{Z}, \forall j}} \sum_{j=1}^{M} \alpha_{j,\lambda^*}(\Delta_j(t), d_j(t), l_j(t)), \tag{5.60}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} l_j(t) \le N. \tag{5.61}$$

The "Net Gain Maximization" problem (5.60) with constraint (5.61) is a bounded Knapsack problem. By using (5.60)-(5.61), we propose a new algorithm for the problem (5.44)-(5.45) in Algorithm 5.

Algorithm 5 Net Gain Maximization Policy

1: Input: Optimal dual variable λ^* obtained in Algorithm 4.
2: Compute $\alpha_{j,\lambda^*}(\delta, d, l)$ using (5.59) for all j, δ, d, l.
3: **for** each time $t \geq 0$ **do**
4: Update $\Delta_j(t)$ and $d_j(t)$ using (5.42) and (5.43) for all source j.
5: Compute $(l_j(t))_{j=1}^M$ by solving problem (5.60)-(5.61).
6: $(b_j(t))_{j=1}^M \leftarrow (\hat{b}_{j,\lambda^*}(l_j(t)))_{j=1}^M$ by using (5.54).
7: **end for**

Algorithm 5 starts from $t = 0$. At time $t = 0$, the algorithm takes the dual variable (transmission cost) λ^* from Algorithm 4 which is run offline before $t = 0$. The "Net Gain" $\alpha_{j,\lambda^*}(\delta, d, l)$ is precomputed for every source j, every feature length l, and every state (δ, d) such that $\delta \in \mathbb{Z}^+$, $l, d \in \{1, 2, \ldots, B_j\}$, where we approximate infinite set of AoI values $\mathbb{Z}^+$ by using an upper bound δ_{bound}. We can set $\alpha_{j,\lambda^*}(\delta, d, l) = \alpha_{j,\lambda^*}(\delta_{\text{bound}}, d, l)$ if $\delta > \delta_{\text{bound}}$.

From time $t \geq 0$, Algorithm 5 solves the knapsack problem (5.60)-(5.61) at every time slot t. The knapsack problem is solved by using a dynamic programming method in $O(MNB)$ time [**?**], where M is the number of sources, N is the number of channels, and B is the maximum buffer size among all source j. The feature position $b_j(t)$ is obtained from a look up table that stores the value of function $\hat{b}_{j,\lambda^*}(l)$ for all j and l.

Unlike the Whittle index policy [33], our policy proposed in Algorithm 5 does not need to satisfy any indexability condition. There exists some other policies that do not need to satisfy indexability condition [86, 88]. The policies in [86, 88] are developed based on linear programming formulations, our policy does not need to solve any linear programming.

5.7 Data Driven Evaluations

In this section, we demonstrate the performance of our scheduling policies. The performance evaluation is conducted using an inference error function obtained from a channel state information (CSI) prediction experiment. In Fig. 5.2, one can observe the inference error function of a CSI prediction experiment. The discrete-time autocorrelation function of the generated fading channel coefficient is defined as $r(k) = bJ_0(2\pi f_d T_s |k|)$, where $r(k)$ represents the autocorrelation of the CSI signal process with time lag k, b signifies the variance of the process, $J_0(\cdot)$ denotes the zeroth-order Bessel function, T_s is the channel sampling

duration, $f_d = \frac{vf_c}{c}$ is the maximum Doppler shift, v stands for the velocity of the source, f_c is the carrier frequency, and c represents the speed of light. In this experiment, we employed a quadratic loss function. Although we utilize the CSI prediction experiment and a quadratic loss function for evaluating the performance of our scheduling policies, we note that our scheduling policies are not limited to any specific experiment, loss function, or predictor.

Single Source Scheduling Policies

We evaluate the following four single source scheduling policies.

1. Generate-at-Will, Zero Wait with Feature Length l: In this policy, $S_{i+1} = S_i + T_i(l_i)$, $b_i = 0$, and $l_i = l$ for all i-th feature transmissions.

2. Optimal Policy with Time-invariant Feature Length (TIFL): The policy that we propose in Theorem 5.1.

3. Optimal Policy with Time-variant Feature Length (TVFL): The policy that we propose in Theorem 5.2.

4. Periodic Updating with Feature Length l: After every time slot T_p, the policy submits features with feature length l and feature position 0 to a First-Come, First-Served communication channel.

We evaluate the performance of the above four single source scheduling policies, where the task to infer the current CSI of a source by observing features. For generating the CSI dataset, we set $b_0 = 1$, $T_s = 1$ms, $v = 15$ m/s, and $f_c = 2$GHz. Additionally, we add white noise to the feature variable with a variance of 10^{-6}.

In the single source case, we consider that the i-th feature requires $T_i(l) = \lceil \alpha l \rceil$ time-slots for transmission, where α represents the communication capacity of the channel. For example, if the number of bits used for representing a CSI symbol is n and the bit rate of the channel is ρ, then $\alpha = \frac{\rho}{n}$.

Fig. 5.5 shows the time-averaged inference error under different policies against the parameter α, where $\alpha > 0$. The plot is constrained to $\alpha = 1$ since values of $\alpha > 1$ is impractical due to the possibility of sending CSI using fewer bits. The buffer size of the

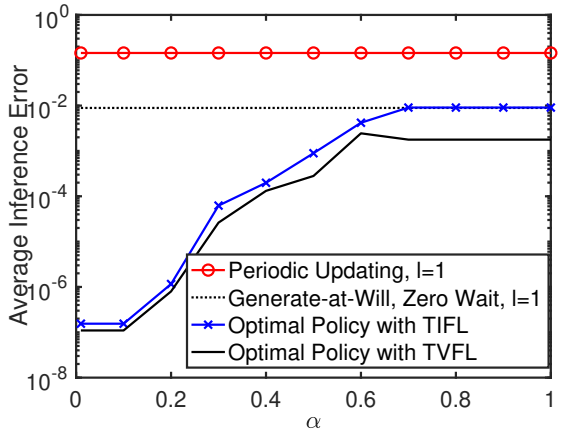


Figure 5.5: Single Source Case: Time-averaged inference error vs. the scale parameter α in transmission time $T_i(l) = \lceil \alpha l \rceil$ for all i.

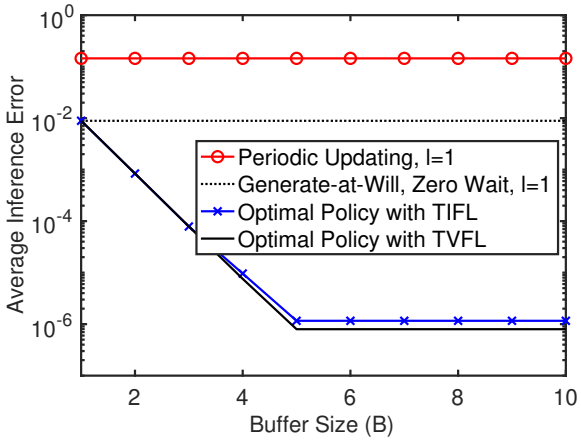


Figure 5.6: Single Source Case: Time-averaged inference error vs. the buffer size B.

source is $B = 10$. Among the four scheduling policies, the "Optimal Policy with TVFL" yields the best performance, while the "Optimal Policy with TIFL" outperforms the other two policies. The findings in Figure 5.5 demonstrate that when $\alpha \leq 0.1$, the "Optimal Policy with TVFL" can achieve a performance improvement of 10^4 times compared to the "Periodic Updating, $l = 1$" with $T_p = 4$ and "Generate-at-Will, Zero Wait, $l = 1$" policies. This result is not surprising since "Periodic Updating, $l = 1$" and "Generate-at-Will, Zero Wait, $l = 1$" do not utilize longer features, despite all features with $l = 1, 2, \ldots, 10$ taking only 1 time slot when $\alpha \leq 0.1$. When $\alpha > 0.1$, the average inference error of the "Periodic Updating" and "Generate-at-Will, Zero Wait" policies are at least 10 times worse than that of the "Optimal Policy with TVFL." The reasons are as follows: (1) The "Periodic Updating" policy does not transmit a feature even when the channel is available, leading to an inefficient use of resources. In our simulation, this situation is evident as $T_i(1) = 1$ and $T_p = 4$. Again, "Periodic Updating" may transmit features even when the preceding feature has not yet been delivered, resulting in an extended waiting time for the queued feature. This frequently leads to the receiver receiving a feature with a significantly large AoI value, which is not good for accurate inference. (2) Conversely, the "Generate-at-Will, Zero-Wait" policy isn't superior because zero-wait is not advantageous, and the feature position $b = 0$ may not be an optimal choice since the inference error is non monotonic with respect to AoI.

The policy "Optimal Policy with TIFL" achieves an average inference error very close to that of the "Optimal Policy with TVFL," but it is simpler to implement. Furthermore, the "Optimal Policy with TIFL" requires only one predictor associated with the optimal time-invariant feature length and does not require switching the predictor.

Fig. 5.6 plots the time-averaged inference error vs. the buffer size B. In this simulation, $\alpha = 0.2$ is considered. The results show that increasing B can improve the performance of the "Optimal Policy with TVFL" and "Optimal Policy with TIFL" compared to the other policies. As B increases, "Optimal Policy with TVFL" and "Optimal Policy with TIFL" outperform the others. In contrast, the "Periodic Updating" and "Generate-at-Will" policies do not utilize the buffer and their performance remains unchanged with increasing B. Moreover, we can notice that the buffer size $B = 5$ is enough for this experiment as further increase in buffer size does not improve the performance.

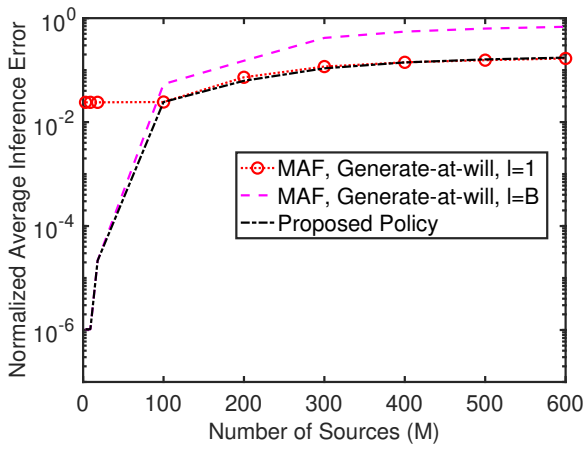


Figure 5.7: Multiple Source Case: Time-averaged inference error vs. the number of sources M.

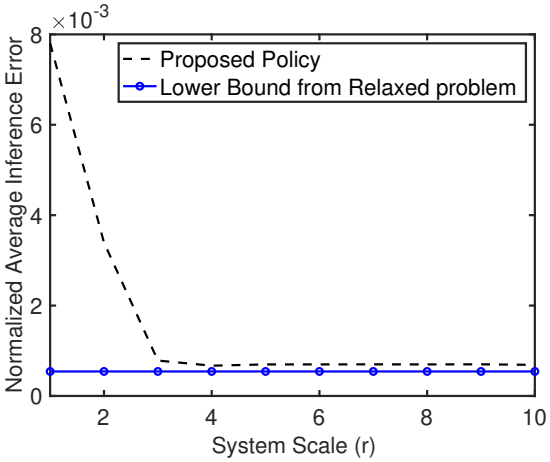


Figure 5.8: Multiple Source Case: Time-averaged inference error vs. system scale r, where $M = 3r$ and $N = 10r$.

Multiple Source Scheduling Policies

In this section, we evaluate the time-averaged inference error of the following three multiple source scheduling policies.

1. Maximum Age First (MAF), Generate-at-will, $l = 1$: As the name suggests, this policy selects the sources with maximum AoI value at each time. Specifically, under this policy, $\min\{N, M\}$ sources with maximum AoI are selected. Moreover, the feature length and the feature position of the selected sources are 1 and 0, respectively.

2. Maximum Age First (MAF), Generate-at-will, $l = B$: This policy also selects the sources with maximum AoI values at each time, but with feature length $l = B$. Under this policy, $\min\{\lfloor \frac{N}{B} \rfloor, M\}$ sources with maximum AoI are selected, where B is the buffer size of all sources, i.e., $B_j = B$ for all source j. Moreover, the feature position of the selected sources is 0.

3. Proposed Policy: The policy in Algorithm 5.

The performance of three multiple source scheduling policies is illustrated in Fig. 5.7, where each source sends its observed CSI to the corresponding predictor. In this simulation, three types of sources are considered: (i) type 1 source with a velocity of $v_1 = 15$ m/s and a CSI variance of $b_1 = 0.5$, (ii) type 2 sources with a velocity of $v_2 = 20$ m/s and a CSI variance of $b_2 = 0.1$, and (iii) type 3 sources with a velocity of $v_3 = 25$ m/s and a CSI variance of $b_3 = 1$.

Fig. 5.7 illustrates the normalized average inference error (the total time-averaged inference error divided by the number of sources) plotted against the number of sources M, with $N = 100$ and $B = 10$. We can observe from Fig. 5.7 that when the number of sources is less, the normalized average inference error of our proposed policy is 10^4 times better than "MAF, Generate-at-will, $l = 1$." However, "MAF, Generate-at-will, $l = B$" is close to the proposed policy. But, When number of sources is more than 400, the normalized average inference error becomes 4 times lower than that of the "MAF, Generate-at-will, $l = B$" policy. As the number of sources increases, the normalized average inference error obtained by "MAF, Generate-at-will, $l = 1$" becomes close to the normalized average inference error of the proposed policy.

Fig. 5.8 compares the time-averaged inference error of the proposed policy and a lower bound from a relaxed problem. The lower bound is achieved by selecting feature length and feature position by using (5.53) and (5.55), respectively under dual cost $\lambda = \lambda^*$ obtained from Algorithm 4. For this evaluation, we have taken step size $10^{-4}/(kr)$ at each iteration k In Algorithm 4. In Fig. 5.8, we consider $N = 10r$ channels and $M = 3r$ sources, where r represents the system scale. Observing Fig. 5.8, it becomes evident that our proposed policy converges towards the lower bound as the system scale r increases.

Appendix

5.A Experimental Setup for ML Experiments in Chapter 5.2

In the first machine learning (ML) experiment: wireless channel state information (CSI) prediction, our objective is to infer the CSI of a source at time t by observing a feature consisting of a sequence of stale and noisy CSIs. Specifically, we consider a Rayleigh fading-based CSI. The dataset is generated by using the Jakes model [91]. In the Jakes fading channel model, the CSI can be expressed as a Gaussian random process. Due to the joint Gaussian distribution of the target and feature random variables, the optimal inference error performance is achieved by a linear MMSE estimator. Hence, a linear regression algorithm is adopted in our simulation. Nonetheless, our study can be readily applied to other neural network-based predictors.

In the second experiment: actuator state prediction, we employ a neural network based predictor. In this experiment, we use an OpenAI CartPole-v1 task [2] to generate a dataset, where a DQN reinforcement learning algorithm [61] is utilized to control the force on a cart and keep the pole attached to the cart from falling over. Our goal is to predict the pole angle at time t based on a sequence of stale information of cart velocity with length l. The predictor in this experiment is an LSTM neural network that consists of one input layer, one hidden layer with 64 LSTM cells, and a fully connected output layer. Additional experiments can be found in Chapter 3, including (a) video prediction and (b) robot state prediction.

5.B Proof of Lemma 5.1

Part (a): Consider the sequence $\tilde{X}_{-\delta}^{l} = (\tilde{V}_{-\delta}, \tilde{V}_{-\delta-1}, \ldots, \tilde{V}_{-\delta-l+1})$. It can be demonstrated that for any $1 \leq l_1 \leq l_2$, the Markov chain $\tilde{Y}_0 \leftrightarrow \tilde{X}_{-\delta}^{l_2} \leftrightarrow \tilde{X}_{-\delta}^{l_1}$ holds. This is due to the fact that for $1 \leq l_1 < l_2$, the sequence $\tilde{X}_{-\delta}^{l_2} = (\tilde{V}_{-\delta}, \tilde{V}_{-\delta-1}, \ldots, \tilde{V}_{-\delta-l_1+1}, \tilde{V}_{-\delta-l_1}, \ldots, V_{-\delta-l_2+1})$ includes $\tilde{X}_{-\delta}^{l_1} = (\tilde{V}_{-\delta}, \tilde{V}_{-\delta-1}, \ldots, \tilde{V}_{-\delta-l_1+1})$ as well as $(\tilde{V}_{-\delta-l_1}, \ldots, \tilde{V}_{-\delta-l_2+1})$. By applying the data processing inequality [48, Lemma 12.1] for L-conditional entropy, we can deduce that

$$H_L(\tilde{Y}_0|\tilde{X}_{-\delta}^{l_2}) \leq H_L(\tilde{Y}_0|\tilde{X}_{-\delta}^{l_1}). \tag{5.62}$$

Part (b): Assuming that (5.10) holds for all $l = 1, 2, \ldots$ and $x \in \mathcal{V}^l$, and employing [12, Lemma 3], [14] yields

$$H_L(P_{Y_t|X^l_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^l_{-\delta}}|P_{X^l_{t-\delta}}) = H_L(Y_t|X^l_{t-\delta}) + O(\beta). \tag{5.63}$$

Combining (5.63) with (5.62), we deduce that

$$\begin{aligned} H_L(P_{Y_t|X^{l_2}_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^{l_2}_{-\delta}}|P_{X^{l_2}_{t-\delta}}) &= H_L(Y_t|X^{l_2}_{t-\delta}) + O(\beta) \\ &\leq H_L(Y_t|X^{l_1}_{t-\delta}) + O(\beta) \\ &= H_L(P_{Y_t|X^{l_1}_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^{l_1}_{-\delta}}|P_{X^{l_1}_{t-\delta}}) + O(\beta) + O(\beta) \\ &= H_L(P_{Y_t|X^{l_1}_{t-\delta}}; P_{\tilde{Y}_0|\tilde{X}^{l_1}_{-\delta}}|P_{X^{l_1}_{t-\delta}}) + O(\beta). \end{aligned} \tag{5.64}$$

This concludes the proof.

5.C Proof of Theorem 5.1

Optimal Feature Length Determination: To find the optimal feature length for the time-invariant scheduling problem (5.13), we undertake a two-step process:

1. *Calculation of $\bar{p}_l$:* Given a feature length l, we start by determining $\bar{p}_l$, defined as

$$\bar{p}_l = \inf_{\pi \in \Pi_l} \limsup_{T \to \infty} \frac{1}{T} \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \mathrm{err}_{\mathrm{inference}}(\Delta(t), l) \right], \tag{5.65}$$

   where Π_l represents the set of admissible policies for feature length l. This step quantifies the optimal objective value for each specific feature length.

2. *Optimal Feature Length and Objective Value:* Having obtained $\bar{p}_l$ for all relevant l, the optimal feature length l^* can be determined by solving

$$l^* = \operatorname*{argmin}_{l \in \mathbb{Z}: 1 \leq l \leq B} \bar{p}_l, \tag{5.66}$$

where B represents an upper bound on the feature length. Additionally, the optimal objective value is given by

$$\bar{p}_{\text{inv}} = \min_{l \in \mathbb{Z}: 1 \leq l \leq B} \bar{p}_l. \tag{5.67}$$

These steps collectively identify the most suitable feature length and its corresponding objective value.

We aim to solve the problem (5.13) by addressing the sub problems (5.65)-(5.66). Let's begin by solving (5.65) using [12, Theorem 4.2], restated below for completeness.

Theorem 5.3 *[12, Theorem 4.2] If the transmission times $T_i(l)$'s are i.i.d. with a finite mean $\mathbb{E}[T_i(l)]$, then there exists an optimal solution $\pi_l^* = ((S_1^*(l), b_1^*(l), l), (S_2^*(l), b_2^*(l), l), \ldots) \in \Pi_l$ to* (5.65) *that satisfies:*

(a) The optimal feature position in π_l^* is time-invariant, i.e., $b_1^*(l) = b_2^*(l) = \cdots = b^*(l)$. The optimal feature position $b^*(l)$ in π_l^* is given by

$$b^*(l) = \underset{0 \leq b \leq B-l}{\operatorname{argmin}} \beta_{b,l}, \tag{5.68}$$

where $\beta_{b,l}$ is the unique root of equation (5.16)*.*

(b) The optimal scheduling time $S_{i+1}^*(l)$ in π_l^* is determined by

$$S_{i+1}^*(l) = \min_{t \in \mathbb{Z}} \left\{ t \geq S_i^*(l) + T_i(l) : \gamma_l(\Delta_b(t), l) \geq \bar{p}_l \right\}, \tag{5.69}$$

where $\Delta_b(t) = t - S_i^*(l) + b^*(l)$ is the AoI at time t. The optimal objective value $\bar{p}_l$ of (5.65) *is*

$$\bar{p}_l = \min_{0 \leq b \leq B-l} \beta_{b,l}. \tag{5.70}$$

Using Theorem 5.3, we obtain values of $\bar{p}_l$ for all $l = 1, 2, \ldots, B$. We can then determine l^* and $\bar{p}_{\text{inv}}$ using (5.66) and (5.67), respectively. Substituting l^* and $\bar{p}_{\text{inv}}$ into the policy $\pi_{l^*}^*$

established in Theorem 5.3, we derive the optimal policy π^*, as asserted in Theorem 5.C. This completes the proof.

5.D Proof of Theorem 5.2

The infinite time-horizon average cost problem (5.26) can be cast as an average cost semi-Markov decision process (SMDP) [69, 72]. To describe the SMDP, we define decision times, action, state, state transition, and cost of the SMDP.

Decision Times and Waiting Time

Each i-th feature delivery time $D_i = S_i + T_i(l_i)$ is considered a decision time. Let Z_{i+1} denote the waiting time between the i-th feature delivery time D_i and the $(i+1)$-th feature sending time S_{i+1}, given by:

$$Z_{i+1} = S_{i+1} - D_i. \tag{5.71}$$

With $S_0 = 0$, we can express $S_{i+1} = \sum_{k=0}^{i} T_k(l_k) + Z_{k+1}$. Thus, given $(T_0, T_1, \ldots)$, we can uniquely determine $(S_1, S_2, \ldots)$ from $(Z_1, Z_2, \ldots)$. Consequently, a policy in Π can be represented as $\pi = ((Z_1, b_1, l_1), (Z_2, b_2, l_2), \ldots)$, where at time D_i, $(Z_{i+1}, b_{i+1}, l_{i+1})$ represents the action.

State and State Transition

At time D_i, the state is $(\Delta(D_i), d(D_i))$. The AoI process $\Delta(t)$ evolves as:

$$\Delta(t) = \begin{cases} T_i(l_i) + b_i, & \text{if } t = D_i, \quad i = 0, 1, \ldots, \\ \Delta(t-1) + 1, & \text{otherwise.} \end{cases} \tag{5.72}$$

The feature length $d(t)$ evolves as:

$$d(t) = \Big\{ l_i, \quad \text{if } D_i \leq t < D_{i+1}. \tag{5.73}$$

Hence, at the decision time D_i, the state value is $(\Delta(D_i), d(D_i)) = (T_i(l_i) + b_i, l_i)$.

Cost

The expected time between two decision times, D_i and D_{i+1}, is given by:

$$\mathbb{E}[D_{i+1} - D_i] = \mathbb{E}[Z_{i+1} + T_{i+1}(l_{i+1})]. \tag{5.74}$$

Given $\Delta(D_i) = \delta$ and $d(D_i) = d$, the expected cost within the interval $[D_i, D_{i+1})$ is:

$$\mathbb{E}\left[\sum_{t=D_i}^{D_{i+1}-1} \mathrm{err}_{\text{inference}}\Big(\Delta(t), d(t)\Big)\right] = \mathbb{E}\left[\sum_{k=0}^{Z_{i+1}+T_{i+1}(l_{i+1})-1} \mathrm{err}_{\text{inference}}\Big(\delta + k, d\Big)\right]. \tag{5.75}$$

Solution via Dynamic Programming: To solve the SMDP, we employ the dynamic programming method [69, 72]. Initially, we define a σ-field and a stopping time set for the state process $(\Delta(t), d(t))$.

Define σ-field

$$\mathcal{F}_s^t = \sigma((\Delta(t+k), d(t+k)) : k \in \{0, 1, \ldots, s\}), \tag{5.76}$$

which is the set of events whose occurrence are determined by the realization of the process $\{(\Delta(t+k), d(t+k)) : k \in \{0, 1, \ldots, s\}\}$ from time slot t up to time slot $t + s$. Then, $\{\mathcal{F}_k^t, k \in \{0, 1, \ldots\}\}$ is the filtration of the process $\{S(t+k) : k \in \{0, 1, \ldots\}\}$. We define $\mathfrak{M}$ as the set of all stopping times by

$$\mathfrak{M} = \{\nu \geq 0 : \{\nu = k\} \in \mathcal{F}_k^t, k \in \{0, 1, 2, \ldots\}\}. \tag{5.77}$$

Given $(\Delta(D_i), d(D_i)) = (\delta, d)$, the optimal action $(Z^*_{i+1}, l^*_{i+1}, b^*_{i+1})$ satisfies the following Bellman optimality equation for the SMDP (5.26):

$$\begin{aligned} h(\delta, d) = \min_{\substack{Z\in\mathfrak{M} \\ l\in\mathbb{Z}:1\le l\le B \\ b\in\mathbb{Z}:0\le b\le B-l}} \Bigg\{ & \mathbb{E}\left[\sum_{k=0}^{Z+T_{i+1}(l)-1} \mathrm{err}_{\mathrm{inference}}(\delta+k, d)\right] \\ & - \mathbb{E}[Z + T_{i+1}(l)]\ \bar{p}_{\mathrm{opt}} + \mathbb{E}[h(T_{i+1}(l)+b, l)]\Bigg\} \\ = \min_{\substack{Z\in\mathfrak{M} \\ l\in\mathbb{Z}:1\le l\le B \\ b\in\mathbb{Z}:0\le b\le B-l}} \Bigg\{ & \mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1} \mathrm{err}_{\mathrm{inference}}(\delta+k, d)\right] \\ & - \mathbb{E}[Z + T_1(l)]\ \bar{p}_{\mathrm{opt}} + \mathbb{E}[h(T_1(l)+b, l)]\Bigg\}, \end{aligned} \tag{5.78}$$

where $\mathfrak{M}$ is the set of stopping times defined in (5.77), and the last equality holds because $T_i(l)$'s are independent and identically distributed.

The Bellman optimality equation (5.78) is complex due to the need to jointly optimize three parameters: feature length l, feature position b, and waiting time Z. If $Z_l(\delta, d)$ defined in (5.33) represents the optimal waiting time for a given feature length l, then equation (5.78) can be simplified as follows:

$$\begin{aligned} h(\delta, d) = \min_{\substack{l\in\mathbb{Z} \\ 1\le l\le B}} \Bigg\{ & \mathbb{E}\left[\sum_{k=0}^{Z_l(\delta,d)+T_1(l)-1} \left(\mathrm{err}_{\mathrm{inference}}(\delta+k, d) - \bar{p}_{\mathrm{opt}}\right)\right] \\ & + \min_{\substack{b\in\mathbb{Z} \\ 0\le b\le B-l}} \mathbb{E}[h(T_1(l)+b, l)]\Bigg\}, \end{aligned} \tag{5.79}$$

which leads to (5.32) and (5.34).

Now, we need to prove that $Z_l(\delta, d)$ is the optimal waiting time for a given feature length l. This is true if $Z_l(\delta, d)$ is the optimal solution to the following optimization problem:

$$\min_{Z\in\mathfrak{M}} \mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1} \left(\mathrm{err}_{\mathrm{inference}}(\delta+k, d) - \bar{p}_{opt}\right)\right]. \tag{5.80}$$

Simplification of the Problem (5.80) The problem (5.80) poses a challenge due to its nature as an optimal stopping time problem. However, we can simplify the problem by exploiting a property of the state transition. Let $\nu^* \in \mathfrak{M}$ represent the optimal stopping time that solves (5.80). For any $k \leq \nu^*$, it holds that $\Delta(D_i+k) = \Delta(D_i)+k$ and $d(D_i+k) = d(D_i)$. Consequently, the set $\{(\Delta(D_i+k), d(D_i+k)) : k = 1, 2, \ldots, \nu^*\}$ is entirely determined by the initial value $(\Delta(D_i), d(D_i))$. Additionally, for all $k \leq \nu^*$, the σ-field $\mathcal{F}_k^{D_i}$ can be simplified as $\mathcal{F}_k^{D_i} = \sigma((\Delta(D_i), d(D_i))$. Thus, any stopping time in $\mathfrak{M}$ corresponds to a deterministic time. As a result, problem (5.80) can be further simplified into the following integer optimization problem:

$$\min_{Z\in\{0,1,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{\mathrm{opt}}\right)\right]. \tag{5.81}$$

We aim to demonstrate that $Z_l(\delta, d)$ is the optimal solution for (5.81).

Optimal Waiting Time Determination: By utilizing (5.81), we can determine that waiting time $Z = 0$ is optimal if the following inequality holds:

$$\begin{aligned}&\min_{Z\in\{1,2,\ldots\}} \mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{\mathrm{opt}}\right)\right]\\ &\geq \mathbb{E}\left[\sum_{k=0}^{T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{\mathrm{opt}}\right)\right].\end{aligned} \tag{5.82}$$

In scenarios where $Z = 0$ is not optimal, the optimal choice becomes $Z = 1$ if the following condition is satisfied:

$$\begin{aligned}&\min_{Z\in\{2,3\ldots\}} \mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{\mathrm{opt}}\right)\right]\\ &\geq \mathbb{E}\left[\sum_{k=0}^{1+T_1(l)-1}\left(\mathrm{err}_{\mathrm{inference}}(\delta+k,d)-\bar{p}_{\mathrm{opt}}\right)\right].\end{aligned} \tag{5.83}$$

Following a similar argument, if $Z = 0, 1, \ldots, \tau - 1$ are not optimal, then $Z = \tau$ becomes optimal when

$$\min_{Z \in \{\tau+1, \tau+2 \ldots\}} \mathbb{E}\left[\sum_{k=0}^{Z+T_1(l)-1} \left(\mathrm{err}_{\mathrm{inference}}(\delta + k, d) - \bar{p}_{opt}\right)\right]$$
$$\geq \mathbb{E}\left[\sum_{k=0}^{\tau+T_1(l)-1} \left(\mathrm{err}_{\mathrm{inference}}(\delta + k, d) - \bar{p}_{opt}\right)\right]. \tag{5.84}$$

Hence, we deduce that the optimal waiting time is the least integer value τ that satisfies (5.84). This inequality can be equivalently expressed as

$$\min_{j \in \{1,2 \ldots\}} \mathbb{E}\left[\sum_{k=0}^{j-1} \left(\mathrm{err}_{\mathrm{inference}}(\delta + \tau + j + T_1(l), d) - \bar{p}_{opt}\right)\right] \geq 0. \tag{5.85}$$

Similar to Lemma 7 in [25], the following lemma holds.

Lemma 5.2 *The following inequality holds*

$$\min_{j \in \{1,2 \ldots\}} \mathbb{E}\left[\sum_{k=0}^{j-1} \left(\mathrm{err}_{\mathrm{inference}}(\delta + \tau + j + T_1(l), d) - \bar{p}_{opt}\right)\right] \geq 0, \tag{5.86}$$

if and only if

$$\min_{j \in \{1,2,\ldots\}} \frac{1}{j} \sum_{k=0}^{j-1} \mathbb{E}\left[\mathrm{err}_{\mathrm{inference}}\left(\delta + \tau + j + T_1(l), d\right)\right] \geq \bar{p}_{opt}. \tag{5.87}$$

The left hand side of (5.87) is exactly $\gamma_l(\delta + \tau, d)$ defined in (5.14).

To conclude the proof, the optimal waiting time corresponds to the least integer value τ that satisfies $\gamma_l(\delta + \tau, d) \geq \bar{p}_{opt}$. This optimal waiting time leads to (5.33).

Chapter 6

Concluding Remarks and Future Works

Next-generation communications (Next-G), such as 6G, are expected to enhance performance, coverage, capacity, and energy efficiency by orders of magnitude compared to existing cellular infrastructures. This objective accompanies the demand for networked intelligent applications including remote monitoring, control, and inference over networks. For the increasing numbers of the networked intelligent applications, the scalability of the network architecture requires a paradigm shift in the communication strategies—from maximizing data transmission rates to extracting and transmitting the right piece of information (often referred to as *semantic information*) to accomplish specific tasks such as accurate inference (e.g., the location of vehicles) or control (e.g., navigation) [14–19, 34]. We will refer to this paradigm as *Semantic* communication. Recent activities in the communication and control communities illustrate the value of such an approach in reducing communication requirements for a given level of application performance [18, 19]. In this dissertation, by considering "Freshness" as a "Semantic" property of information, we focus on the design of communication systems to improve the performance of a remote inference system.

Summary of Contributions

In **Chapter 1**, we introduce a remote inference system and its applications for networked intelligence. Moreover, we define Age of Information (AoI) as the measure of information freshness in a remote inference system. We found inference error can be expressed as a function of AoI. We use inference error as the performance metric and AoI as the semantic metric to design timely communication protocols. In **Chapter 2**, we interpret how to evaluate the importance of fresh information in a remote inference system by using a new information-theoretic tool called ϵ-Markov chain. We found that fresher data is always better for Markovian data sequences, but fresher data is NOT necessarily better for non-Markovian data sequences. Based on these insights, in **Chapter 3**, we propose a new "Selection-from-Buffer" status updating model that has the option to select the most important feature among the features stored in the source buffer. We design optimal selection-from-buffer

scheduling policies for single-source and single-channel remote inference networks. Furthermore, in **Chapter 4**, we design an asymptotically optimal selection-from-buffer scheduling policy for multiple-source and multiple-channel remote inference networks by integrating the Whittle index and a duality-based feature selection algorithm. In **Chapter 5**, we analyzed the joint impact of timeliness and feature sequence length in a remote inference system. We develop learning and communications co-designs that jointly control the timeliness and the length of the feature sequences. For a single source-predictor pair and a single channel, we develop optimal co-designs for both time-invariant and time-varying feature lengths. For multiple source-predictor pairs and multiple channels, we develop learning and communication co-design by using a new gain index.

Future Works

The following research problems could be interesting directions based on our findings.

- *How to design a semantic encoder and predictor/controller?*–Sensor measurements can be of high dimensional and can require huge communication resources. Instead of sending high-dimensional sensor measurements, we can design a neural network that extracts low-dimensional information from high-dimensional measurements. In the future, we will jointly optimize a neural network (NN)-based semantic encoder and a NN-based predictor/controller for end-to-end communications. This problem requires to address high-dimensional parameter spaces and dynamic environments.

- *Online scheduling problem for remote inference/control:* When the statistics of the communication channel are unknown, it is required to formulate a robust online learning algorithm. Moreover, an online learning algorithm is necessary for the case of non-stationary data sequences. The structure of our optimal scheduling policies is useful for designing online scheduling algorithms.

- *Reinforcement Learning-based Scheduler*: For the multi-source setting, we need to answer: *which sources to select?* and *what information to send (from the selected sources)?* This scheduling decision is a *multi-action restless multi-armed bandit (RMAB)*

problem [12, 14, 26, 33, 34, 87], which is PSPACE-hard [75]. Our works [12, 14, 34] introduced new asymptotically optimal policies using *gain index* and *Whittle index*. The indices are difficult to compute for high dimensional state space and non-stationary environment [92, 93]. We will design deep *reinforcement learning* (RL)-based scheduling algorithms, which are best-suited for decision problems with high dimensional state space and changing environments.

Bibliography

[1] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv:1804.01523*, 2018.

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv:1606.01540*, 2016.

[3] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.

[5] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6G internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022.

[6] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," 2016, arXiv:1604.07316.

[7] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *IEEE ICRA*, 2018, pp. 3758–3765.

[8] C. She, R. Dong, Z. Gu, Z. Hou, Y. Li, W. Hardjawana, C. Yang, L. Song, and B. Vucetic, "Deep learning for ultra-reliable and low-latency communications in 6G networks," *IEEE network*, vol. 34, no. 5, pp. 219–225, 2020.

[9] X. Song and J. W.-S. Liu, "Performance of multiversion concurrency control algorithms in maintaining temporal consistency," in *Fourteenth Annual International Computer Software and Applications Conference*. IEEE, 1990, pp. 132–133.

[10] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *IEEE INFOCOM*, 2012, pp. 2731–2735.

[11] M. K. C. Shisher, H. Qin, L. Yang, F. Yan, and Y. Sun, "The age of correlated features in supervised learning based forecasting," in *IEEE INFOCOM Age of Information Workshop*, 2021.

[12] M. K. C. Shisher and Y. Sun, "How does data freshness affect real-time supervised learning?" *ACM MobiHoc*, 2022.

[13] ——, "On the monotonicity of information aging," in *IEEE INFOCOM ASoI Workshop*, 2024.

[14] M. K. C. Shisher, Y. Sun, and I.-H. Hou, "Timely communications for remote inference," *submitted*, 2023.

[15] Y. E. Sagduyu, S. Ulukus, and A. Yener, "Task-oriented communications for NextG: End-to-end deep learning and ai security aspects," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 52–60, 2023.

[16] C. Zhang, H. Zou, S. Lasaulce, W. Saad, M. Kountouris, and M. Bennis, "Goal-oriented communications for the IoT and application to data compression," *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 58–63, 2022.

[17] C. Ari, M. K. C. Shisher, E. Uysal, and Y. Sun, "Goal-oriented communications for remote inference with two-way delay," *under review in IEEE ISIT*, 2024.

[18] M. Merluzzi, M. C. Filippou, L. G. Baltar, and E. C. Strinati, "Effective goal-oriented 6G communications: the energy-aware edge inferencing case," in *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2022, pp. 457–462.

[19] Y. E. Sagduyu, T. Erpek, A. Yener, and S. Ulukus, "Multi-receiver task-oriented communications via multi-task deep learning," *arXiv preprint arXiv:2308.06884*, 2023.

[20] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *IEEE ISIT*, 2015, pp. 3008–3012.

[21] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[22] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," in *IEEE INFOCOM*, 2016, pp. 1–9.

[23] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *J. Commun. Netw.*, vol. 21, no. 3, pp. 204–219, 2019.

[24] ——, "Information aging through queues: A mutual information perspective," in *Proc. IEEE SPAWC Workshop*, 2018.

[25] T. Z. Ornee and Y. Sun, "Sampling and remote estimation for the ornstein-uhlenbeck process through queues: Age of information and beyond," *IEEE/ACM Trans. on Netw.*, vol. 29, no. 5, pp. 1962–1975, 2021.

[26] V. Tripathi and E. Modiano, "A Whittle index approach to minimizing functions of age of information," in *IEEE Allerton*, 2019, pp. 1160–1167.

[27] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer, "AoI-penalty minimization for networked control systems with packet loss," in *IEEE INFOCOM Age of Information Workshop*, 2019, pp. 189–196.

[28] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4019–4034, 2021.

[29] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *IEEE INFOCOM*, 2018, pp. 1844–1852.

[30] Y. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," in *ISIT*, 2018, pp. 2634–2638.

[31] J. Sun, Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, "Closed-form Whittle's index-enabled random access for timely status update," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1538–1551, 2019.

[32] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, 2018.

[33] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, vol. 25, no. A, pp. 287–298, 1988.

[34] M. K. C. Shisher, B. Ji, I. Hou, and Y. Sun, "Learning and communications co-design for remote inference systems: Feature length selection and transmission scheduling," *IEEE J. Select. Areas in Inf. Theory*, 2023.

[35] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, "Effect of message transmission path diversity on status age," *IEEE Trans. Inf. Theory*, vol. 62, no. 3, pp. 1360–1374, March 2016.

[36] T. Soleymani, S. Hirche, and J. S. Baras, "Optimal self-driven sampling for estimation based on value of information," in *IEEE WODES*, 2016, pp. 183–188.

[37] G. Chen, S. C. Liew, and Y. Shao, "Uncertainty-of-information scheduling: A restless multiarmed bandit framework," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 6151–6173, 2022.

[38] Z. Wang, M.-A. Badiu, and J. P. Coon, "A framework for characterising the value of information in hidden Markov models," *IEEE Trans. Inf. Theory*, 2022.

[39] T. Z. Ornee and Y. Sun, "A Whittle index policy for the remote estimation of multiple continuous Gauss-Markov processes over parallel channels," *ACM MobiHoc*, 2023.

[40] J. Pan, Y. Sun, and N. B. Shroff, "Sampling for remote estimation of the wiener process over an unreliable channel," *ACM Sigmetrics*, 2023.

[41] Y. Sun and S. Kompella, "Age-optimal multi-flow status updating with errors: A sample-path approach," *J. Commun. Netw.*, vol. 25, no. 5, pp. 570–584, 2023.

[42] Y. Sun, I. Kadota, R. Talak, and E. Modiano, *Age of information: A new metric for information freshness*. Springer Nature, 2022.

[43] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Select. Areas in Commun.*, vol. 39, no. 5, pp. 1183–1210, 2021.

[44] T. Z. Ornee and Y. Sun, "Performance bounds for sampling and remote estimation of gauss-markov processes over a noisy channel with random delay," in *IEEE SPAWC*, 2021.

[45] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the Wiener process for remote estimation over a channel with random delay," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, pp. 1118–1135, 2020.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[47] P. D. Grünwald and A. P. Dawid, "Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory," *Annals of Statistics*, vol. 32, no. 4, pp. 1367–1433, 08 2004.

[48] A. P. Dawid, "Coherent measures of discrepancy, uncertainty and dependence, with applications to Bayesian predictive experimental design," *Technical Report 139*, 1998.

[49] F. Farnia and D. Tse, "A minimax approach to supervised learning," *NIPS*, vol. 29, pp. 4240–4248, 2016.

[50] M. K. C. Shisher, T. Z. Ornee, and Y. Sun, "A local geometric interpretation of feature extraction in deep feedforward neural networks," *arXiv:2202.04632*, 2022.

[51] I. S. Dhillon and J. A. Tropp, "Matrix nearness problems with bregman divergences," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 4, pp. 1120–1146, 2008.

[52] I. Csiszár and P. C. Shields, "Information theory and statistics: A tutorial," 2004.

[53] S.-L. Huang, A. Makur, G. W. Wornell, and L. Zheng, "On universal features for high-dimensional learning and inference," *accepted to Foundations and Trends in Communications and Information Theory: Now Publishers*, 2019, available in arXiv:1911.09105.

[54] Y. Polyanskiy and Y. Wu, "Lecture notes on information theory," *Lecture Notes for MIT (6.441), UIUC (ECE 563), Yale (STAT 664)*, no. 2012-2017, 2014.

[55] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[56] M. Shaked and J. G. Shanthikumar, *Stochastic orders*. Springer Science & Business Media, 2007.

[57] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[58] A. Papoulis and S. Unnikrishna Pillai, *Probability, random variables and stochastic processes (4th edition)*, 2002.

[59] F. Ebert, C. Finn, A. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *CoRL*, 2017.

[60] D. Berenson, S. S. Srinivasa, D. Ferguson, A. Collet, and J. J. Kuffner, "Manipulation planning with workspace goal regions," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 618–624.

[61] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[62] P. Attri, Y. Sharma, K. Takach, Shah, and Falak, "Timeseries forecasting for weather prediction," 2020, online: https://keras.io/examples/timeseries/timeseries_weather_forecasting/.

[63] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650–1662, 2005.

[64] J. Liao, O. Kosut, L. Sankar, and F. P. Calmon, "A tunable measure for information leakage," in *IEEE ISIT*, 2018, pp. 701–705.

[65] X.-D. Zhang, *Matrix analysis and applications*. Cambridge University Press, 2017.

[66] S.-I. Amari, "α -divergence is unique, belonging to both f-divergence and bregman divergence classes," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4925–4931, 2009.

[67] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019, vol. 49.

[68] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, 2003, vol. 1.

[69] D. Bertsekas, *Dynamic programming and optimal control: Volume I*. Athena scientific, 2017, vol. 1.

[70] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[71] M. N. Katehakis and A. F. Veinott Jr, "The multi-armed bandit problem: decomposition and computation," *Mathematics of Operations Research*, vol. 12, no. 2, pp. 262–268, 1987.

[72] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[73] I. M. Verloop, "Asymptotically optimal priority policies for indexable and nonindexable restless bandits," *The Annals of Applied Probability*, vol. 26, no. 4, p. 1947–1995, 2016.

[74] N. Gast, B. Gaujal, and C. Yan, "LP-based policies for restless bandits: necessary and sufficient conditions for (exponentially fast) asymptotic optimality," *arXiv:2106.10067*, 2021.

[75] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queueing network control," in *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*, 1994, pp. 318–322.

[76] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Select. Areas in Commun.*, vol. 24, no. 8, pp. 1439–1451, 2006.

[77] N. Gast, B. Gaujal, and C. Yan, "Exponential asymptotic optimality of whittle index policy," *Queueing Systems*, vol. 104, no. 1, pp. 107–150, 2023.

[78] V. Tripathi, L. Ballotta, L. Carlone, and E. Modiano, "Computation and communication co-design for real-time monitoring and control in multi-agent systems," in *IEEE WiOpt*, 2021, pp. 1–8.

[79] I. Kadota, A. Sinha, and E. Modiano, "Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1359–1372, 2019.

[80] T. Soleymani, J. S. Baras, and K. H. Johansson, "Stochastic control with stale information–part I: Fully observable systems," in *IEEE CDC*, 2019, pp. 4178–4182.

[81] B. Sombabu, A. Mate, D. Manjunath, and S. Moharir, "Whittle index for AoI-aware scheduling," in *IEEE COMSNETS*, 2020, pp. 630–633.

[82] M. Chen, K. Wu, and L. Song, "A Whittle index approach to minimizing age of multi-packet information in IoT network," *IEEE Access*, vol. 9, pp. 31 467–31 480, 2021.

[83] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Optimizing information freshness using low-power status updates via sleep-wake scheduling," in *ACM MobiHoc*, 2020, pp. 51–60.

[84] J. Pan, A. M. Bedewy, Y. Sun, and N. B. Shroff, "Minimizing age of information via scheduling over heterogeneous channels," *ACM MobiHoc*, 2021.

[85] G. Chen and S. C. Liew, "An index policy for minimizing the uncertainty-of-information of Markov sources," *IEEE Trans. Inf. Theory*, 2023.

[86] G. Xiong, X. Qin, B. Li, R. Singh, and J. Li, "Index-aware reinforcement learning for adaptive video streaming at the wireless edge," in *ACM MobiHoc*, 2022, pp. 81–90.

[87] Y. Zou, K. T. Kim, X. Lin, and M. Chiang, "Minimizing age-of-information in heterogeneous multi-channel systems: A new partial-index approach," in *ACM MobiHoc*, 2021, pp. 11–20.

[88] Y. Chen and A. Ephremides, "Scheduling to minimize age of incorrect information with imperfect channel state information," *Entropy*, vol. 23, no. 12, p. 1572, 2021.

[89] D. J. Hodge and K. D. Glazebrook, "On the asymptotic optimality of greedy index heuristics for multi-action restless bandits," *Advances in Applied Probability*, vol. 47, no. 3, pp. 652–667, 2015.

[90] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *Journal of applied probability*, vol. 27, no. 3, pp. 637–648, 1990.

[91] W. C. Jakes and D. C. Cox, *Microwave mobile communications*. Wiley-IEEE press, 1994.

[92] K. Nakhleh, S. Ganji, P.-C. Hsieh, I. Hou, S. Shakkottai *et al.*, "NeurWIN: Neural whittle index network for restless bandits via deep rl," *Advances in Neural Information Processing Systems*, vol. 34, pp. 828–839, 2021.

[93] G. Xiong and J. Li, "Finite-time analysis of whittle index based Q-learning for restless multi-armed bandits with neural network function approximation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

Vita

B.Sc. in Electrical and Electronic Engineering — Feb. 2013 - Sep. 2017
Bangladesh University of Engineering and Technology

Lecturer, Electrical and Electronic Engineering Dept. — Nov. 2017 - June 2018
Bangladesh Army University of Science and Technology

M.S. in Electrical Engineering — Aug. 2022
Auburn University

Graduate Research and Teaching Assistant — Aug. 2018- May. 2024
Auburn University

Publications

1. **Md Kamran Chowdhury Shisher**, Yin Sun, and I-Hong Hou, "Timely Communications for Remote Inference," *submitted*, 2023.

2. Cagri Ari, **Md Kamran Chowdhury Shisher**, Elif Uysal, and Yin Sun, "Goal-Oriented Communications for Remote Inference with Two-Way Delay," *IEEE ISIT*, 2024.

3. **Md Kamran Chowdhury Shisher** and Yin Sun, "On the Monotonicity of Information Aging," *IEEE INFOCOM ASoI Workshop*, 2024.

4. **Md Kamran Chowdhury Shisher**, Bo Ji, I-Hong Hou, and Yin Sun, "Learning and Communications Co-Design for Remote Inference Systems: Feature Length Selection and Transmission Scheduling," *IEEE Journal on Selected Areas in Information Theory*, vol. 4, pp. 524-538, 2023.

5. Kevin Yan, **Md Kamran Chowdhury Shisher**, and Yin Sun, "A Transfer Learning-Based Deep Convolutional Neural Network for Detection of Fusarium Wilt in Banana Crops," *AgriEngineering*, vol. 5, no. 4, pp. 2381-2394, 2023.

6. Tasmeen Zaman Ornee, **Md Kamran Chowdhury Shisher**, Clement Kam, and Yin Sun, "Context-aware Status Updating: Wireless Scheduling for Maximizing Situational Awareness in Safety-critical Systems," *IEEE MILCOM*, 2023.

7. **Md Kamran Chowdhury Shisher** and Yin Sun, "How Does Data Freshness Affect Real-time Supervised Learning?" *ACM MobiHoc*, 2022.

8. **Md Kamran Chowdhury Shisher**, Heyang Qin, Lei Yang, Feng Yan, and Yin Sun, "The Age of Correlated Features in Supervised Learning based Forecasting," *IEEE INFOCOM AoI Workshop*, 2021.